

Universidad
Carlos III de Madrid

Ingeniería Técnica de Informática de Gestión

PROYECTO FIN DE CARRERA

Aplicación de la Herramienta Open Source Sloodle y las Tecnologías del Procesamiento del lenguaje Natural para el Desarrollo de una Plataforma de Virtual Learning en la Universidad Carlos III de Madrid

Autor: Don Eduardo Rojo Sánchez.

Tutor/ Director: Dr. David Griol Barres

Octubre 2010

Título: Aplicación de la Herramienta Open Source Sloodle y las Tecnologías del Procesamiento del lenguaje Natural para el Desarrollo de una Plataforma de Virtual Learning en la Universidad Carlos III de Madrid

Autor: Eduardo Rojo Sánchez

Director: David Griol Barres

EL TRIBUNAL

Presidente: ___________________________________

Vocal: ___________________________________

Secretario: ___________________________________

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _______ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Quería dar gracias ante todo a mi tutor del proyecto, el Dr. David Griol, que es la persona que me ha estado animando hasta el final, y gracias a su profesionalidad a la hora de dirigir y coordinar el proyecto hemos podido desarrollarlo rápidamente.

Me alegro mucho de haber trabajado con David en este proyecto, pues ha hecho que diera lo mejor de mí a la hora de desarrollarlo, consiguiendo como resultado un proyecto con un gran potencial.

Me gustaría dedicar el proyecto a Jesús Conde que fue la persona que me enseño Second Life, y gracias a la cual conseguí un alto nivel a hora de trabajar en los metaversos.

Otra persona a la cual sólo conozco mediante su trabajo es el Sr. Albert Revolution, que gracias a su trabajo sobre OpenSim estoy seguro que conseguirá potenciar toda esta tecnología basada en mundos virtuales.

Resumen

El presente Proyecto Final de Carrera describe cómo desarrollar entornos educativos en los mundos virtuales de Second Life y en los generados mediante OpenSimulator, utilizando herramientas de gran valor como es el caso de Sloodle. Para ello, se ha desarrollado el metaverso en Opensim de la UC3M, con el primordial objetivo de enseñar cómo configurar nuestro propio mundo virtual e instalarlo en un servidor dedicado, conectarlo dentro una página web y consultar la información de una manera más parecida al mundo real.

En primer lugar explicamos los diversos mundos virtuales que existen e indicamos las personas más importantes y sitios de referencia que debemos consultar. En segundo lugar, se explica con detalle cómo configurar las diversas herramientas que ofrece Moodle para la enseñanza virtual o e-learning, habiendo configurado incluso una plataforma Moodle que recoge toda la información del proyecto. En tercer lugar, se detallan las diversas herramientas que ofrece Sloodle para el desarrollo de entornos educativos a distancia, mostrando cómo realizar su conexión con nuestra plataforma Moodle. En último lugar, explicamos la configuración de OpenSimulator, para crear nuestro propio mundo virtual, el intercambio P2P, desarrollar nuestro propio megaverso y poder utilizar las diversas herramientas de Sloodle.

Adicionalmente, hemos desarrollado una aplicación práctica de las Tecnologías del Habla y el Procesamiento del Lenguaje Natural para construir un bot conversacional, capaz de acatar nuestras ordenes y además interactuar mediante la voz para proporcionar respuesta a un conjunto de consultas dentro del ámbito académico. Para el desarrollo de este bot se han utilizado programación en lenguaje C# y tecnologías y aplicaciones como bases de datos mySQL, librerías específicas como libopenmetaverse para la conexión a mundos virtuales, reconocimiento automático del habla mediante Microsoft SAPI y síntesis de voz mediante Loquendo TTS.

El proyecto se ha desarrollado para que no exista ningún inconveniente en aplicarlo comercialmente, ya que proporciona la información necesaria para poder administrar nuestro propio metaverso configurado en un servidor dedicado o virtual y poder integrarlo en diferentes exploradores web, ofreciendo así la posibilidad de ofrecer servicios de administración y creación de mundos virtuales.

Palabras clave: Mundos Virtuales, Metaverso, E-Learning, Second Life, OpenSimulator, Moodle, Sloodle, Bot, Sistemas de Dialogos, Interacción Natural.

Abstract

This Final Project describes how to develop educative applications in virtual worlds, like Second Life or the ones that can be generated by means of OpenSimulator, and using tools of great value as it is the case of Sloodle. To do this, we have develop a metaverse in OpenSim for the UC3M, with the main objective to describe how our own virtual world has been built, to install it in a dedicated server, to connect it using a webpage, and to consult the information of one more a way more similar to the real world.

Firstly, we explain the main virtual worlds that exist and we indicate the most important researchers and reference sites that we have consult. Secondly, the different tools that Moodle offers for virtual education are explained in detail. We have even developed a Moodle platform that collects all the informarion of the project. Thirdly, the se of tools that Sloodle offers for the development of educative environments are descrisbed, showing how to complete their connection to our Moodle platform. Then, we explain the configuration of OpenSimulator, to create our own virtual world, to carry out P2P communications, to develop our own megaverse and to be able to use the tools provided by Sloodle.

In addition, we have developed a practical application of Speech Technologies and Natural Language Processing to construct a conversational boot, which is able to accept orders and interact with us by means of the voice to provide an answer to a set of queries within the academic scope. For the development of this bot we have used programming in C# language, and technologies and applications as mySQL databases, specific libraries like libopenmetaverse for the connection to virtual worlds, automatic speech recognition by means of Microsoft SAPI and voice synthesis by means of Loquendo TTS.

The project has been developed to avoid any disadvantagefor applying it, since it provides the information that is needed to manage our own metaverse in a dedicated or virtual server, and to be able to integrate it in different web browsers, therefore offering the possibility of providing management and virtual world creation.

Keywords: Virtual worlds, Metaverse, E-Learning, Second Life, OpenSimulator, Moodle, Sloodle, Bot, Dialogue Systems, Natural Interaction.

Índice

CAPÍTULO I

INTRODUCCIÓN Y OBJETIVOS

Capítulo 1

Introducción y objetivos

1.1. Introducción

El desarrollo de la denominada Web 2.0 ha hecho posible la introducción de numerosas aplicaciones utilizadas diariamente por un gran número de usuarios. Estas aplicaciones han originado nuevas formas y canales de comunicación que están cambiando profundamente las formas de comunicación entre los usuarios de Internet. Las redes sociales han surgido en este marco como un fenómeno de consumo global durante los últimos años. La relevancia adquirida por este tipo de actividades está favoreciendo la aparición de nuevas formas de comunicarse, de compartir información y de interactuar que nos afectan también en nuestra vida cotidiana.

Con el avance en estas tecnologías, se han producido durante la última década enormes avances en el desarrollo de mundos virtuales o *metaversos*. Estos mundos pueden definirse como entornos gráficos 3D simulados por ordenador "cohabitados" por los usuarios a través de sus avatares.

Tradicionalmente, los mundos virtuales se han estructurado a priori predefiniendo las tareas realizables por los usuarios. En la actualidad, en los mundos sociales virtuales, la interacción social posee un papel clave y los usuarios pueden determinar sus experiencias en el mundo virtual siguiendo sus propias decisiones.

De este modo, los mundos virtuales se han transformado en verdaderas redes sociales útiles para la interacción entre personas de diferentes lugares que pueden socializar, aprender, entretenerse, etc. Debido al potencial social de los mundos virtuales, se han convertido en un atractivo para instituciones, empresas e investigadores, con la finalidad de desarrollar robots virtuales con las mismas apariencias y capacidades que los avatares correspondientes a usuarios humanos. Estos

robots virtuales se denominan “metabots", término acuñado a partir de la contracción de los términos metaverso y robot. Por tanto, un metabot es un software completamente capaz de interactuar en una o más metaversos a través de uno o varios avatares.

La principal motivación de este Proyecto Fin de Carrera es el desarrollo de entornos educativos en estos mundos virtuales, favoreciendo una interacción con el alumno de una manera más humana, proporcionándole una interfaz web gráfica que se asemeja al mundo real, y permitiendo que cualquier alumno desde cualquier sitio pueda recibir clases. De esta manera, los alumnos no han de adaptarse sin condiciones a los requisitos impuestos por las limitaciones del diseño web en 2D y los materiales educativos tradicionales, que les obligan a navegar por páginas web que son poco intuitivas e interactivas, de tal manera que mucha información importante les pasa desapercibida.

Los mundos virtuales se asemejan a nuestra vida diaria, en la cual elegimos entrar un edificio determinado para que se nos proporcione en él una determinada información o servicio específico, consiguiendo está mayor facilidad a la hora de poder encontrar y comunicarnos de forma natural con el personal del centro docente, como por ejemplo, entrar en un edificio y elegir un despacho, conectado una persona al metaverso al igual que actualmente utilizamos programas como el correo electrónico, Messenger o Skype.

Mediante la utilización de e-learning en 3D facilitamos, además de que los alumnos no se tengan que adaptar a las máquinas, una interacción más natural entre los alumnos a distancia y los docentes, obteniendo una enseñanza más social y divertida; ya que los mundos virtuales nos proporcionan la oportunidad de crear juegos divertidos que posibilitan que los alumnos se diviertan a la vez que participan en la clase.

El problema fundamental de la mayoría de construcciones de sitios virtuales dedicados a la enseñanza en la actualidad, es que están diseñados para ser un juguete, siendo construcciones huecas dedicadas exclusivamente a que se vean bonitas, y desaprovechando totalmente las posibilidades que permiten los mundos virtuales.

Los mundos virtuales actualmente están además bajo el control de grandes empresas como Linden Lab, Microsoft, IBM e Intel, que presentan mundos virtuales desérticos, ya que en lo único que se basan es en establecer relaciones comerciales por Internet, lo cual se refleja en que las tierras de estas empresas están llenas de tiendas y también en que nos cobren por prácticamente todo. La evolución continua de la web ha generado la necesidad cada vez mayor de desarrollar mundos virtuales libres, al igual que actualmente hay millones de páginas web que son propiedades de los usuarios, en el futuro los mundos virtuales no serán controlados por grandes empresas a base de talonario, evolucionarán en base al talento de las personas, proporcionando estructuras que no estén huecas, y que hacen que la web sea de todos.

En nuestro proyecto basamos el e-learning 3D tanto en el mundo virtual de Second Life como los generados con OpenSimulator, que es un simulador de código abierto que utiliza el mismo estándar que Second Life, permitiendo de esta manera la utilización de herramientas como Sloodle (Moodle más Second Life), que están desarrolladas para impartir clase de forma profesional y divertida, favoreciendo la sociabilidad entre los alumnos.

Las herramientas de Sloodle se basan en las comunicaciones *xml_rcp*, que consisten en el envío de mensajes entre el paquete CMS de Moodle (web en 2D) y los *prims* (cualquier objeto 3D) de los mundos virtuales de Second Life o OpenSimulator. De esta forma, mediante estas

comunicaciones podemos disponer de un sistema experto en enseñanza, como es Moodle, aplicado a la enseñanza en mundos virtuales. Este tipo de comunicaciones nos permiten conectarnos a estos mundos virtuales mediante lenguajes de programación de alto nivel (PHP, Java, Visual Basic, Lindel Scripting Languaje, C#, etc.), de tal forma que nuestra página web y nuestros programas pueden conectarse e interaccionar con estos mundos virtuales, favoreciendo la integración con toda clase de aplicaciones web y permitiendo el acceso a los recursos incluso desde terminales móviles, como teléfonos o PDAs.

Los metabots intensifican la percepción del mundo virtual, proporcionando gestos, miradas, expresiones faciales y movimientos necesarios para el proceso de comunicación. Sin embargo, la interacción social en mundos virtuales se lleva a cabo generalmente en modo de texto mediante servicios de tipo chat. Por este motivo, adicionalmente nos propusimos enriquecer la comunicación en estos entornos, añadiendo capacidades de conversación a los metabots. Con este fin, proponemos la integración de sistemas de diálogo para la construcción de metabots inteligentes con la capacidad de conversar oralmente y, al mismo tiempo, beneficiarse de las modalidades visuales que proporcionan estos mundos virtuales.

Con este fin, hemos llevado a cabo una aplicación práctica de esta integración mediante el desarrollo de un metabot conversacional que proporciona información académica en el mundo virtual de Second Life. De este modo, el metabot desarrollado puede además beneficiarse de la posibilidad de interactuar gran número de usuarios que ofrecen estos entornos

En este Proyecto Fin de Carrera, con el afán de ir más allá, también explicamos la administración y configuración de OpenSimulator, para poder instalarlo en un servidor dedicado o en su propio ordenador y favorecer de esta manera que cada persona igual que puede tener su página web, también pueda tener su propio mundo virtual (metaverso) libre, que es lo que realmente pensamos que dará lugar a la evolución de los mundos virtuales. Existen además otros simuladores de código abierto basados en OpenSimulator, como es el caso de RealXtend, que añaden módulos dentro de OpenSimulator, de tal manera que se mejoren aún más las funcionalidades gráficas.

1.2. Objetivos

El Proyecto Fin de carrera realiza un estudio detallado de las principales características y herramientas para llevar a cabo aplicaciones educativas en mundos virtuales basados en Second Life y OpenSimulator.

Hemos clasificado los objetivos del proyecto según hagan referencia de aplicaciones desarrolladas o manuales para documentar el trabajo desarrollado.

- **Aplicaciones desarrolladas:**
    - **Creación de un sitio Moodle en internet donde está disponible toda la documentación del proyecto.**

      En la siguiente dirección web se puede acceder a toda la documentación y programas que utilizamos en nuestro proyecto:

      http://tesis.edumoot.com/

En este sitio web, además de la documentación del proyecto, se puede encontrar:

- Enlaces a las Sim (porciones de territorio) de Second Life, para así acceder a ellos a través de un curso Moodle.
- Videos de las pruebas realizadas.
- Código fuente ya compilado y preparado para descargar.
- Acceso a programas que hemos utilizado.
- Creación de enlaces desde los cursos de Moodle para acceder desde ellos a varias islas.

- **Creación de nuestro propio mundo virtual mediante OpenSimulator**.

  En nuestra computadora hemos instalado el programa OpenSimulator y gracias a él hemos podido crear nuestro propio mundo virtual, con funcionalidades similares a las de Second Life y de forma completamente gratuita.

- **Conexión de nuestro metaverso a otro metaverso mediante Hypergrid, con la reación de un grid de OpenSim, el grid OsUC3MGrid.**

  Un *grid* no es más que la unión de la potencia y recursos suministrados por varios ordenadores para la creación así de un metaverso de mayor tamaño o megaverso.

- **Conexión de nuestra isla de OpenSim a OsGrid.**

  OsGrid es el único mundo virtual gratuito que existe y se están desarrollando actualmente las herramientas de Sloodle para él. Por lo tanto, este mundo virtual es sumamente importante, ya que gracias a él podremos tener infinitas islas conectadas de forma gratuita.

- **Desarrollo de un bot conversacional que interactúa automáticamente con los otros avatares.**

  Este bot ofrece las siguientes funcionalidades:
  - Responde a preguntas específicas definidas dentro del dominio académico.
  - Se conecta a una base de datos.
  - Tiene la capacidad de hablar. Utilizamos nuestra voz y texto para comunicarnos con el bot, y éste nos responde hablando y mediante texto.

        - Se conecta a los mundos virtuales Second Life y OsGrid.
        - Es capaz de enviar mensajes (IM).
        - Obedece nuestras órdenes.

- **Desarrollo de manuales y entornos de documentación:**
    - **Desarrollo de un manual para la herramienta SCRATCH, que nos permite crear de forma muy fácil y sencilla scripts para Second Life.**
    - **Desarrollo de un manual detallado en un entorno web para cada una de las herramientas de Sloodle.**

1.3. Fases del desarrollo

1.3.1. Planificación y análisis

En la primera reunión del proyecto se definieron los objetivos principales del mismo, ya que antes de aceptar el proyecto había que delimitarlo. Debido a mis altos conocimientos en Second Life, todas las ideas de cómo realizar el proyecto las tenía en mente, excepto el apartado de la realización de un bot que se conectara a Second Life para responder a preguntar realizadas por una persona.

Todo el mérito del apartado del bot, es de mi tutor, David Griol Barres, pues él fue la persona que tuvo la idea de utilizar una librería para conectarse a Second Life, y así manejar el bot. Tras utilizar la librería de LibOpenMetaverse para controlar el bot, el Proyecto Fin de Carrera estaba totalmente completado, pues todas las ideas estaban en mi mente.

La parte fundamental del proyecto, se desarrolló en mes y medio y lo que hay llevado más tiempo ha sido el apartado de los detalles con respecto a la depuración de la memoria y la realización del bot, que le hemos dedicado cinco meses y medio.

Por tanto, la planificación y análisis del Proyecto Fin de Carrera, se realizo en una reunión, y los detalles sobre el desarrollo del proyecto los llevamos a cabo mediante la comunicación por medio de correos electrónicos.

1.3.2. Diseño

Diseño del sitio Web

Toda la memoria y los programas necesarios para desarrollar el proyecto se han dispuesto en nuestra plataforma Moodle específicamente creada para este proyecto, de tal manera que cualquier persona pueda acceder a toda la información del proyecto mediante Internet. Hemos utilizado las herramientas que Moodle, las cuales nos ofrecen los servicios de foro, wiki, diccionario de conceptos, para cada uno de los apartados en los que se divide el proyecto.

Diseño del metaverso y del grid de la UC3M en OpenSimulator

Hemos desarrollado toda la documentación necesaria para que cualquier persona o centro pueda crear en muy poco tiempo su propio metaverso o su propio grid de opensim, y que incluso

pueda configurarlo en un servidor dedicado y, de esta manera, disponer de un terreno o sim que poder administrar.

Diseño de la base de datos

La aplicación del bot conversacional, ha obligado a la creación de una base de datos MySQL, desarrollada mediante un diseño entidad/relación para que el bot pudiera responder automáticamente a las preguntas realizadas por los usuarios sobre información relativa a la universidad.

Diseño del curso Audio Visual de OpenSim

Hemos desarrollado un Curso AudioVisual sobre OpenSim, para facilitar el aprendizaje de estas tecnologías, y de esta manera, dar a conocer cómo configurar nuestro propio metaverso y fomentar así la creación de mundos virtuales personales.

1.3.2. Desarrollo funcional

OpenSimulator, está suficientemente maduro para ser utilizado, ya que compañías como Microsoft, Intel y Gloogle, han creado sus propios mundos virtuales. Para el desarrollo de nuestro bot automático, se ha tenido especial consideración en las pruebas realizadas para la conexión a distintos metaversos como Second Life y Opensimulator, la comprobación que acataba nuestras ordenes, en el funcionamiento de la base de datos a la hora de hacer preguntas que requirieran la selección de bastante información para responder a la pregunta y en que el bot pudiera mover los labios, mover las manos y hablara cuando respondiera una pregunta, para darnos la sensación de que hablábamos con una persona.

En el apartado del bot, hemos utilizado el lenguaje C# para poder programar las clases necesarias para conectarnos a la base de datos MySql, y para poder utilizar el API de Windows Vista; y de esta manera extraer la información de la base de datos para podemos responder a las preguntas realizadas por cualquier avatar.

Se construyó un archivo ejecutable, que se instala en el ordenador para la utilización de este autómata en cualquier ordenador con Windows Vista y con la voz de Loquendo en Español instalada en su ordenador.

1.4. Medios empleados

Lenguajes de Programación

El lenguaje utilizado en este proyecto fin de carrera ha sido C#, que se ha utilizado primero para conectar el bot a los mundos virtuales de Second Life y OpenSimulator mediante la librería libopenmetaverse, para conectar a la base de datos consiguiendo los datos necesarios para las consultas planteadas y, por último, para utilizar el SAPI5 consiguiendo que el bot pudiera hablar.

Se ha utilizado MySql para la creación y gestión de la base de datos necesaria para los datos que hemos hecho uso para que el bot pudiera responder a las preguntas planteadas.

Adicionalmente, se han empleado los lenguajes LSL (Lindel Scripting Lab), PHP y Java para poder realizar las comunicaciones mediante *xml_rcp* a los *prims*. De esta manera, podemos

comunicar una página web o un programa en Java a los mundos virtuales de Second Life y Opensim.

Programas empleados

El principal programa utilizado es Visual Studio con Microsoft .Net Framework 2.0, que se utiliza para generar todo el código fuente que utilizaremos en el proyecto.

El servidor web WampServer, que integra las tecnologías de Apache, PHP y MySql, para que nuestra computadora se convierta en un servidor web, y nos permite conectar el bot, a nuestra base de datos MySql y también lo usamos para utilizar el lenguaje de programación de PHP y de esta manera conectarnos a los prims de Second Life y OpenSim.

El SAPI5 de Windows Vista, que es el encargado de controlar las propiedades de voz, que convierte el texto que escribimos en una voz, y de esta manera el bot habla.

El programa de reconocimiento de voz de Windows Vista que lo utilizamos para convertir nuestra voz en texto, y así interactuar con la interfaz gráfica de los visores de los mundos virtuales para comunicarnos con el bot.

El programa Opensim binaries, que nos permite crear un metaverso en nuestro propio ordenador o un megametaverso con la unión de varios ordenadores. Sin embargo, el futuro de los mundos virtuales esta en el intercambio p2p, que utiliza hypergrid, y que nos permite enlazar metaverso.

Manuales

La documentación examinada consta esencialmente de materiales extraídos de páginas web y blogs de internet dedicados a Second Life, OpenSimulator, Sloodle, Moodle y LibOpenMetaverse. Sin embargo, los sitios principales consultados han sido wiki.secondlife.com, opensimulator.org, openmetaverse.org, moodle.org y hispagrid.com.

1.5. Esquema de la memoria

Tras este capítulo de introducción, el resto de la memoria se estructura en los siguientes capítulos:

- **Capítulo II: Estado del Arte**

  En este capítulo se describen las tecnologías y trabajos más importantes para el desarrollo de entornos educativos en mundos virtuales, así como de las características y utilidades fundamentales del mundo virtual Second Life y los generados con OpenSimulator. Además, se realiza un resumen de la evolución de dichos mundos y tecnologías

- **Capítulo III. Moodle.**

  Realizamos un estudio sobre Moodle, presentando las diversas utilidades que consta para luego poder utilizarlas con Sloodle.

Hemos creado una plataforma en Moodle alojada en la red donde está disponible todo el material del proyecto (código fuente, videos, enlaces,…)

- **Capítulo IV. Second Life**

Realizamos un estudio de las características principales que posee Second Life y de las utilidades necesarias para crear un entorno educativo en este mundo virtual.

- **Capítulo V. Sloodle.**

Llevamos a cabo un estudio de Sloodle (integración de Second Life y Moodle) que permite dar a conocer las múltiples ventajas que ofrece está tecnología para el desarrollo de entornos de v-learning.

Realizamos un análisis de Sloodle y de su aplicación práctica de las diversas utilidades (registro de los avatares en la plataforma Moodle, registro en los cursos que determine el administrador, interconexión del chat de Moodle con el chat de Second Life, herramientas para la realización de cuestionarios, glosarios, incorporación de gestos del avatar, acceso a blogs, uso de salas de conferencias) para el desarrollo de un entorno educativo.

- **Capítulo VI. Bot.**

Las redes sociales, y más concretamente los mundos virtuales, suponen un escenario perfecto para llevar a cabo estas líneas de investigación. En este capítulo presentamos un estudio relativo a la integración de sistemas de diálogo en redes sociales, describiendo en nuestro trabajo el desarrollo de un sistema de diálogo que proporciona información académica para crear un avatar conversacional dentro del mundo virtual de Second Life. Además, el bot desarrollado es capaz de acatar las órdenes que le comuniquemos mediante la voz y el texto.

- **Capítulo VII. OpenSim.**

Realizamos un estudio de OpenSim, creando nuestro propio metaverso gratuito en nuestro propio ordenador, y haciendo accesible nuestro metaverso desde internet para que cualquier avatar se pueda conectar a nuestro metaverso UC3M.

Hemos conectado nuestro mundo virtual en OpenSim a OsGrid, que es un grid de OpenSim gratuito, y de esta manera conseguimos infinititas islas gratuitas para cada curso, para poder dar clase en un metaverso.

Deatallamos también la herramienta hypergrid que nos permite crear enlaces a otros metaversos o grid, para poder teletransportar un avatar de uno a otro metaverso al cual hayamos hecho un hypergrid. Es el mismo concepto que se usa en las páginas web, al hacer un enlace de una página web de un servidor web a la página web de otro servidor web, cuando hacemos clic en el enlace nos vamos a otra página web.

- **Capítulo VIII. Conclusiones y trabajo futuro**

  En este capítulo se resumen las principales conclusiones del estudio detallado en capítulos anteriores. Asimismo, se refleja también el grado de consecución de los objetivos propuestos. Por último, se analizan las posibilidades de cara al futuro que tiene las diferentes tecnologías, además de las posibles mejoras que se podrían llevar a cabo.

- **Glosario**

- **Bibliografía**

CAPÍTULO II

ESTADO DEL ARTE

Capítulo 2

Estado del Arte

2.1. *Introducción*

Los metaversos son entornos donde los humanos interactúan social y económicamente en un ciberespacio que es una abstracción del mundo real, pero en el cual las leyes físicas desaparecen o pueden crear sus propias leyes físicas.

Un punto importante es el relativo a la accesibilidad de los metaversos, para conseguir que las personas que tengan alguna minusvalía puedan interactuar libremente en ellos. Para lograr este objetivo hay actualmente visores *open source* (código abierto), lo que significa que los usuarios pueden mejorarlo y añadir funcionalidades a la interfaz de usuario, además de distintos dispositivos de entrada que faciliten dicha interacción. Un ejemplo actual fue el patch para el ratón 3D SpaceNavigator [64], que está actualmente integrado en el visor original de Linden Lab.

Otro punto importante en los metaversos es la posibilidad de poder comunicar los *prims*[1] internos del metaverso con el mundo externo, enviando mensajes a un objeto dentro de una región, siempre y cuando éste se haya registrado para recibir dicha información. Por lo tanto, se puede enviar datos al *prim* utilizando un lenguaje de programación como Java o PHP, lo que nos posibilita poder conectar los *prim* a la tecnología web existente. Un ejemplo de estas comunicaciones es Sloodle [27], que es la unión de Moodle [5] y Second Life [1], mediante la

---

[1] En Second Life se utiliza el término "prim" para referirse a los objetos primitivos. Excepto los avatares, el suelo del simulador y los árboles, todos los objetos en Second Life están hechos de Prims. De esta forma, se utilizan los términos "vestido Prim" o "pelo Prim" o "zapatos Prim" para referirse a objetos hechos de prims.

comunicación de los *prims* con las páginas web de Moodle programadas mediante el lenguaje de programación PHP.

2.2. Trabajos relacionados con nuestro proyecto

La mejor manera para recopilar información más actual, para obtener los conceptos y las teorías fundamentales sobre Moodle, Second Life, Sloodle, LibOpenMetaverse [29] y OpenSim [11] son sus páginas web, sus respectivos foros y en sus wikis. También cabe destacar las páginas web personales de algunos usuarios expertos que desinteresadamente ofrecen información de muy alta calidad, y muchas veces mejor explicadas que en los propios sitios web, y con ideas que no se comentan en la página web de los desarrolladores de la herramienta.

2.2.1. Trabajos importantes en Second Life

El mundo virtual de Second Life desde el principio no me ha supuesto ningún problema para trabajar con él, pues la razón por la que escogí realizar este proyecto era porque mis conocimientos respecto al mundo virtual de Second Life y las herramientas utilizadas en él eran bastante altos. Estos conocimientos de Second Life me sirvieron tanto para poder desarrollar todo el proyecto, como para comprender perfectamente los artículos consultados en relación con él.

Para consultar información sobre Second Life deberemos ir a su wiki [33], que contiene por ejemplo el material necesario para poder aprender sobre el scripting en Second Life [34]. Una nueva característica importante de Second Life es la configuración Lip Sync [35] del navegador de Second Life, que ha sido deshabilitada por defecto ya que es una versión beta. Con esta funcionalidad podemos hacer que nuestro avatar mueva los labios mientras hablamos por el micrófono en el navegador de Linden Lab y, de esta manera, conferir un aspecto mucho más real a nuestro avatar. Esta configuración de Lip Sync la podemos aprovechar para la creación de los bot, para hacer que mueva los labios, y así nos dará la sensación de que hablamos con una persona.

Para desarrollar el apartado de Second Life, la sección de Scratch, he utilizado el video tutorial de Jesús Eduardo Conde.

2.2.2. Trabajos importantes en Moodle

La herramienta Moodle posee en su propia página web [36] libros gratuitos en español que explican de forma muy didáctica y fácil cómo aprender a usar la herramienta de enseñanza online tanto para profesores como para alumnos. Por lo tanto, puede consultarse en ella el conocimiento más importante sobre esta herramienta de una forma muy sencilla, consiguiendo un gran nivel en muy poco tiempo.

Uno de los principales problemas que cabe resolver al utilizar Sloodle es la necesidad de disponer de un sitio Moodle. La primera solución que pensé fue utilizar la dirección IP de una máquina de la Universidad pero aquí surge el problema de disponer de cortafuegos que impiden que desde otra máquina se pudiera conectar a un sitio Moodle de nuestra propia máquina, al no disponer de permisos de administrador de la red.

Para solucionar este problema, pensamos en instalar en un servidor gratuito el paquete Moodle y así poder utilizarlo, pero la idea fracasó al no poder alojar todos los archivos mediante FTP en

los servidores Moodle gratuitos, y además no se proporcionan servicios de mantenimiento ni ninguna garantía. Adicionalmente, debe tenerse en cuenta que es mucho más difícil configurar el paquete Moodle en un servidor web, que en nuestro propio ordenador de manera local.

Un proyecto que hay que destacar con respecto a Moodle es el proyecto *Key to School* (KTS) [7], el cual hemos utilizado tanto para la elaboración y supervisión de la memoria de este proyecto como para llevar a cabo las pruebas con Sloodle en este proyecto. *Key to School* es un hosting gratuito de Moodle, que no nos introduce publicidad como otros hosting, y en el que se actualiza periódicamente a la última versión de Moodle y Sloodle.

Durante todo el desarrollo del proyecto nunca hemos tenido ningún problema con este servidor, la única salvedad que nos puede presentar es que no podemos acceder a la base de datos de nuestro sitio Moodle, pues no tenemos permiso para acceder a ella, y tampoco tenemos los permisos necesarios para aumentar el tamaño máximo de archivos que se puede alojar en este sitio Moodle, es decir, no podemos subir un archivo que su tamaño sea superior a 128 megabyte, al contrario que pasaría si fuéramos nosotros los propietarios de nuestro sitio Moodle.

En *Key To School* creen que Moodle es un regalo para todos los educadores, ya que, observaron que la creación y el mantenimiento de un servidor Moodle es una ardua tarea para muchos educadores debido a las restricciones impuestas por muchos proveedores de alojamiento no administrado. De este modo, se han esforzado para facilitar el aprendizaje y proporcionar el conjunto de recursos necesarios de una forma completamente gratuita. Se esfuerzan, por tanto, para facilitar el espacio en disco y cuota de ancho de banda necesarios para que los profesores puedan llevar a cabo la docencia presencial y poseen una política de "ningún anuncio" para poder crear actividades sin distracciones para los estudiantes.

Para desarrollar el apartado de Moodle he utilizado el libro de Jesús Baños Sancho [10], que es una persona destacada con respecto a Moodle, gracias al manual de referencia desarrollado sobre la utilización de esta herramienta. Claro está que se trata de una gran ayuda, pero aún así hay detalles importantes de Moodle que no están reflejados en el manual, como por ejemplo trucos a la hora de crear los cursos, también cómo organizar en una carpeta maestra toda la información para tener toda la información agrupada en un mismo lugar, etc. Por tanto, aunque sea un manual muy bien explicado y sea un manual de referencia, el trabajo de aprendizaje no termina nunca, sino que siempre hay que estar investigando y viendo distintos puntos de vista, de esta manera nos daremos cuenta del gran potencial que presentan muchas personas a la hora de realizar su trabajo.

Un ejemplo de lo que hablo soy yo mismo, a pesar de haber terminado el proyecto con un nivel bastante alto de calidad y una alta preparación, puedo leer los foros que contienen nuevas ideas, que me dejan maravillado y a la vez me animan a investigar y probar cosas nuevas, y me doy cuenta que aunque mi proyecto lleve un nivel de preparación bastante alto, nunca terminaré de aprender, por lo tanto nunca se puede llegar a cerrar ningún proyecto, ningún manual, pues todo está continuamente evolucionando. Sirva con ello para destacar la importancia del apartado de estado del arte de esta memoria, pues es en él de mencionan personas excepcionales que son precisamente las que permiten esta evolución, pues están constantemente ayudando e investigando.

2.2.3. Trabajos importantes en Sloodle

Para desarrollar el apartado de Sloodle, he utilizado la página oficial de Sloodle donde dan explicaciones de cómo configurar las herramientas, en la cual aparecía video tutoriales realizados por Jesús Eduardo Conde, los cuales son un trabajo excepcional, propio de un maestro. Dentro del propio Second Life, en la Sim de Sloodle se imparte clases por si se tiene alguna duda sobre las herramientas de Sloodle.

Las herramientas sobre Sloodle están continuamente actualizándose, siendo una verdadera lástima que la documentación asociada no sea del todo clara y completa. Muchas veces la información de cómo configurar las herramientas de Sloodle viene adjunta en las propias herramientas de Sloodle. La mayor parte de la documentación está diseñada por desarrolladores, lo cual no es malo del todo, pues una vez que llevas trabajando bastante tiempo con estas herramientas y has estado informándote y leyendo varias veces la documentación, has leído en foros, has consultado en wikis, es justamente ésta la ayuda que se necesita. Para poder seguir aprendiendo sobre las nuevas herramientas que van creando en Sloodle y acerca de cómo configurarlas un buen sitio donde acudir es la documentación de Sloodle [37].

El trabajo que se incluye en este proyecto sobre Sloodle explica con el máximo detalle cada una de las herramientas, su configuración y ejemplos de utilización. Este trabajo ha requerido una investigación exhaustiva y validación de las diferentes funcionalidades mediante experiencias prácticas.

Una persona destacada con respecto a Sloodle es el profesor Mike McKay [40], que dispone de cursos audiovisuales accesibles en YouTube describiendo la forma de configurar y utilizar las diversas herramientas que presenta Sloodle. También ha creado una escuela dentro del mundo virtual de Second Life en la isla de Sloodle, donde es posible matricularse a las clases y solicitar también la resolución de dudas de forma completamente gratuita.

2.2.3. Trabajos importantes en desarrollo de bots

Para desarrollar el apartado en la memoria del bot, he utilizado la librería LibOpenMetaverse, que posee una wiki que enseña el código fuente necesario para crear un bot básico, y también posee un API bastante detallado. Esta Librería es realmente útil para conectarnos a cualquier metaverso basado en Second Life y OpenSim, sin embargo, es bastante difícil de utilizar y existe muy poca documentación en internet sobre cómo utilizarla. El foro de LibOpenMetaverse [38] es, por tanto, un lugar de gran utilidad dónde preguntar dudas, y podemos conocer la respuesta a problemas que han tenido personas que anteriormente han utilizando esta herramienta.

El software OpenSim es totalmente configurable para crear un metaverso o un grid. El mejor sitio para informarse sobre su configuración es su propia página web, pero cabe tener en cuenta que debido a la gran cantidad de datos que presenta es difícil hacerse una idea de lo que realmente puede hacerse y existen numerosas funcionalidades con poca o nula documentación al respecto.

La importancia del trabajo desarrollado en el proyecto final de carrera radica en combinar las funcionalidades más importantes de Opensim (como es creación de un metaverso, crear un grid, crear un hypergrid, física ninja, etc.) y su fusión con Sloodle, presentando diversos ejemplos de funcionamiento. También se incluye una explicación de diversos metaversos virtuales como Osgrid, RealXtend y ReactionGrid.

La información sobre LibOpenMetaverse es realmente escasa, siendo el mejor sitio para consultar información su propia wiki [39]. Por supuesto el API de LibOpenMetaverse es esencial a la hora de programar, pues es la única manera de saber los métodos que van asociados a cada evento. Hay disponibles también listas para desarrolladores en las que pueden consultarse dudas, pero realmente el número de personas subscritas es muy pequeño.

El bot PikkuBot en SecondLife [42] es uno de los muchos proyectos que hemos visto que trabajan con la librería LibOpenMetaverse. Todos estos proyectos tienen en común que al bot se le suministran órdenes para que las acate, acerca de las cuales pueden obtenerse una buena idea de cómo programar nuestro bot y apreciar los servicios que puede ofrecer.

El bot Ariel [43] es un prototipo de asistente académico basado en Inteligencia Artificial, capaz de interactuar con los alumnos mediante lenguaje natural de una forma bastante limitada, debido a que la iniciativa del diálogo es completamente llevada a cabo por el sistema y el número de funcionalidades proporcionadas es muy limitado.

Para incluir la funcionalidad de reconocimiento automático del habla en nuestro bot, hemos utilizado el Speech Recognition API incluido en el sistema operativo Windows Vista [41], que permite la creación de gramáticas en las que poder incluir el modelo de lenguaje necesario para el reconocedor.

2.2.4. Trabajos importantes en OpenSim

Una de los websites que destaca de forma excepcional dada su visión bastante amplia con respecto a los metaversos es el de Albert Revolution [44]. En su trabajo se puede consultarse información detallada sobre Second Life, OpenSim, OsGrid, OpenLife, visores basados en metaversos, RealXtend y creación de objetos en 3D.

Sin lugar a dudas el trabajo realizado es realmente bueno, pues muestra siempre los avances que existen con respecto a OpenSim, incorpora manuales sobre su configuración, sobre cómo crear objetos en 3D para desarrollar nuestros metaversos, cómo configurar diversas utilidades para los metaversos como VideoTrace (a partir de una grabación en video crea un objeto en 3D para mundos virtuales que refleja luz, sombras y superficies), facilita el contacto con personas punteras en la creación de metaversos, incluye críticas detalladas sobre el estado actual de los metaversos libres, informa de las vulnerabilidades al entrar en los grid de OpenSim, etc. Gracias a los artículos publicados por Albert Revolution podemos, por tanto, darnos cuenta de la situación actual de OpenSim y conocer todo el contexto que rodea al desarrollo de mundos virtuales.

Para desarrollar el apartado en la memoria de OpenSim, y su sección OsGrid, se ha utilizado principalmente el manual que hay en la página web de Grupo la Cripta, integrado por Albert Revolution, Darok Kaminski, Leizah Rosana, Silvia Mills Alesandra Capra. También está basado este apartado en una adaptación a este manual que está desarrollado por el maestro Jesús Eduardo Conde.

El proyecto HispaGrid [45], facilita además un foro donde se incluye información muy valiosa y puntera sobre la configuración de OpenSim y RealXtend. La información que se presenta en este foro es de gran calidad y difícil de encontrar en otras fuentes en la red, lo que demuestra el gran nivel que poseen las personas que son capaces de escribir tales artículos. Por tanto, debe considerarse como un punto de referencia para acudir y para de aprender sobre estas materias.

Entre los manuales de gran calidad que se incluyen en este portal cabe destacar "*Voice en OpenSim*" [65], que explica como incorporar la voz en OpenSim. Se trata de un artículo de una gran calidad en cuanto a explicación, utilidad y nivel, pues demuestra Delaier Saenz un nivel de configuración de OpenSim realmente alto.

También cabe destacar el artículo de Albert Revolution sobre cómo abrir nuestra SIM al exterior, que explica la herramienta dyndns para conseguir una IP pública fija, y de esta manera cada vez que nos conectemos a internet con una IP dinámica no sea necesario cambiar la IP en la configuración de OpenSim. El artículo escrito también por Albert Revolution acerca de cómo restaurar la base de datos de la aplicación es también muy valioso, pues nos ayuda a restablecer el sistema de forma segura sin que se produzca la pérdida de la información de la base de datos.

En la tecnología RealXtend [46] cabe destacar que se está desarrollando actualmente un gran trabajo mejorar su aspecto gráfico y de forma totalmente gratuita, aunque muchas veces una

funcionalidad superior no significa que las personas lo acepten, pues gran parte de los usuarios que están acostumbrados a Second Life no quieren algo mejor, sino lo mismo pero de forma gratuita. Éste puede considerarse como uno de los retos que debe afrontar RealXtend para ampliar su uso. De nuevo, cabe destacar el manual desarrollado por Albert Revolution, detallando las mejoras que ofrece RealXtend respecto a OpenSim con respecto a su mejor diseño gráfico la utilización de herramientas de modelado en 3D para objetos (al contrario de lo que sucede con Second Life, donde para crear cualquier objeto es necesario utilizar las nada cómodas herramientas que los desarrolladores de Second Life ofrecen).

Para aprender a configurar Realxtend lo mejor es aprender primero a configurar openSim. Ciertamente la documentación que puede encontrarse en su página web es el único lugar donde aprender a configurar la herramienta, mediante la lectura de importantes ficheros como *readme_rexserverplatform.txt*.

El problema fundamental de la documentación de Realxtend es que está anticuada, especialmente para la creación de grids, y no ofrece la posibilidad de foros donde poder suministrar consultas. Por supuesto, cabe destacar que se están desarrollando nuevos visores que nos permiten teletransportar nuestro avatar y nuestro inventario a otros metaversos, y además configurar el terreno de una SIM. Realmente se trata de un proyecto muy fuerte y que puede llegar a tener un gran potencial por la calidad de su trabajo.

Otro website que cabe destacarse es el blog de *Marlonj* [47], dedicado a explicar diversos metaversos y proporcionar información e ideas de cómo configurar metaversos y diversas herramientas para Second Life. La principal limitación es la desestructuración de la información presentada.

Otro portal a destacar es el de Irene Muni [48], el cual esta realmente muy bien hecho, con una gran cantidad de manuales que nos ayudan a tener nuevas ideas y poder consultar las dudas fundamentales que tengamos sobre Second Life y OpenSim. Dispone de foros para ayudar a las personas que están interesadas en crear su propio metaverso en OpenSim.

La documentación realizado por Atlante Guerrero [49] es otra de las referencias que hay que tener en cuenta sobre la materia. Destaca por la buena organización de su documentación y su forma brillante de explicar y redactar, así como por el buen material que ofrece a los desarrolladores. En él podemos localizar fácil y ordenadamente cualquier manual y su explicación, imágenes que se introducen, y claridad con que se muestran cómo crear objetos en SecondLife, scripting y SIM hacen que el lector se sienta muy cómodo a la hora de trabajar con el excelente material que comparte. En su trabajo se puede ver también cómo expresa su valiosa opinión sobre Second Life, consultar explicaciones sobre el uso de herramientas en este mundo virtual, participar en cursos audiovisuales, aprender sobre las tecnologías RealXtend y OpenSim, etc. En definitiva, se trata de un gran aporte a la comunidad GNU.

2.3. Análisis de las alternativas de los mundos virtuales

El mundo virtual OpenLife [50] es un grid basado en OpenSim. OpenLife ha configurado un grid de OpenSim y ha puesto a alquilar sus terrenos, hay que tener en cuenta que los terrenos en un metaverso tienen un coste en cuanto a compra y mantenimiento de servidores.

El mundo virtual Wonderland [51] ha sido creado por Sun Microsystems. Ofrece un conjunto de herramientas desarrolladas totalmente en Java para crear mundos virtuales colaborativos en 3D, siendo por tanto multiplataforma. Su objetivo principal está centrado en el trabajo colaborativo para posibilitar que sus usuarios pueden comunicarse con los avatares en alta fidelidad, audio inmersivo, compartir el escritorio, aplicaciones y documentos en tiempo real, negociar, etc. No es un entorno centralizado, y al tener su código bajo licencia GPL las empresas pueden utilizarlo para desarrollar su propio mundo virtual para interactuar con sus clientes.

El mundo virtual Croquet [52] es una poderosa tecnología de software de código abierto. Tiene como objetivo permitir la colaboración entre usuarios, actuando como un sistema de conferencia online construido sobre una interfaz 3D y una arquitectura de red P2P. Puede considerarse como una especie de sistema operativo en línea, con aplicaciones compartidas con otros usuarios y con una arquitectura abierta.

Croquet integra una alta gama de sonido tridimensional y gráfico, y tecnologías de colaboración, simulaciones y otras herramientas para crear un espacio para compartir, y con un gran alcance para la creación, uso y reutilización de contenidos digitales. Creo que este proyecto es importante por tres razones: por lo que ofrece, por la forma en que puede ampliar nuestras ideas sobre la nueva información, la colaboración y los ambientes de aprendizaje, y por lo que puede enseñarnos acerca de cómo diseñar entornos de software nuevos e innovadores que se apoyen, en lugar de competir directamente con la infraestructura de información existente. Croquet representa un gran paso en nuestra visión de la computación como una plataforma de comunicaciones y el servicio, disponible en cualquier momento y en cualquier lugar, desde cualquier dispositivo. (por ejemplo, PDAs).

El mundo virtual Open Cobalt [53] pone a disposición de todas las personas una plataforma de código libre y abierto para la construcción, el acceso y la participación en espacios de trabajos virtuales para la investigación y la educación. Open Cobalt hace posible que las personas hipervinculen sus mundos virtuales en 3D a través de portales para formar una gran red distribuida de espacios de colaboración entre sí. También hace posible que las escuelas y otras organizaciones fijen libremente sus propias redes de espacios de trabajo público y privado virtual en 3D, ofreciendo navegación integrada web, chat de voz, chat de texto, y acceso a aplicaciones de escritorio remoto.

Open Cobalt ha sido desarrollado con la principal funcionalidad de un desarrollo sencillo para que particulares y organizaciones creen un mundo virtual seguro. Al aprovechar los gráficos 3D basados en OpenGL, soporta la visualización de colaboración altamente escalable de datos, aprendizaje virtual y desarrollo de entornos de resolución de problemas, wikis en 3D, juegos online, así como la privacidad y la seguridad en entornos virtuales que facilitan la comunicación de grupo. A través de la capacidad de Open Cobalt, se puede disponer de todos sus recursos en red (sistemas de aprendizaje de gestión, sistemas de gestión de recursos, wikis, etc) en sus espacios virtuales de forma interactiva. Open Cobalt trata de estimular el desarrollo y toma de decisiones en colaboración entre todos los usuarios, sí como la búsqueda y resolución de problemas entre los miembros de las organizaciones virtuales distribuidas y comunidades educativas.

El mundo virtual 3rd Rock Grid [54] es un grid basado en OpenSim, y consiste en una comunidad social de individuos en todo el mundo que interactúan en un mundo virtual en 3D. El contenido, la apariencia y presentación del mundo son creadas por sus propios usuarios. En este mundo virtual te encuentras con casas, empresas, clubes de música, zonas de juego, parques, museos, es decir, cualquier tipo de desarrollos que decida construir su comunidad. También se puede conocer y hablar con otras personas de distintas partes del mundo.

El mundo virtual NewWorld [55] es un mundo virtual francés, abierto a todas las etnias, y centrado en los seres humanos, fundado en los valores humanos como la libertad, el respeto y la compartición de recursos. Este aspecto es el que posibilita que los mundos virtuales tengan un corportamiento más interactivo, social, dinámico, humano, lo que les hace más accesible a todos, y en particular mediante el apoyo a proyectos educativos y culturales con los mundos virtuales como medio de comunicación y como plataforma de colaboración.

La organización sin fines de lucro Virtus [66], gestiona el NewWorld [55], recaudando fondos para hacerla evolucionar gracias a sus usuarios, del voluntariado y del alquiler de tierras, para apoyar así activamente este tipo de proyectos.

El mundo virtual Open Neuland y Wilder Westen [56] es un mundo alemán 3D creado por los desarrolladores de mundos virtuales TalentRaspel Ltd. Está basado en el software de código abierto OpenSimulator. El mundo virtual virtyou [57] fue fundado en 2009. Virtyou GmbH es una empresa alemana centrada en la convergencia de medios en los mundos 3D interactivos. El mundo virtual Your Alternative Life [58] es un mundo virtual multiusuario que permite a sus usuarios interactuar entre sí a través de avatares en 3D, y se puede crear y comerciar con propiedades virtuales y servicios.

CAPÍTULO III

MOODLE

Capítulo 3

Moodle

3.1 Introducción

Tal y como se ha comentado en el capítulo segundo de esta memoria, Moodle es un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista.

Moodle se distribuye gratuitamente como Software libre (Open Source) (bajo la Licencia Pública GNU). Básicamente esto significa que Moodle tiene derechos de autor (copyright), pero que se dispone de ciertas libertades. De hecho, puede copiarse, usarse y modificarse siempre que se acepte: proporcionar el código fuente a otros, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado de él. Moodle puede funcionar en cualquier ordenador en el que esté instalado PHP, y soporta varios tipos de bases de datos (en especial MySQL).

La palabra Moodle se corresponde con el acrónimo de Modular Object-Oriented Dynamic Learning Environment (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular), características que resultan fundamentalmente útiles para programadores y teóricos de la educación. También es un verbo que describe el proceso de deambular perezosamente a través de algo, y hacer las cosas cuando se te ocurre hacerlas. Estas dos acepciones pueden aplicarse tanto a la manera en que se desarrolló Moodle como a la metodología en la que un estudiante o profesor podría aproximarse al estudio o enseñanza de un curso en línea.

El trabajo que se resume en este capítulo muestra de manera sencilla y eficaz las explicaciones necesarias para poder llevar a cabo la construcción de un sitio Moodle donde crear cursos mediante las configuraciones de las diversas utilidades que nos presenta esta herramienta, para luego poder incluso combinar estas herramientas con las que nos ofrece Sloodle, y así poder construir un entorno educativo más completo.

3.2. Instalación de Moodle de forma local

3.2.1. Descargar Moodle

En este apartado describimos cómo descargar Moodle en nuestro ordenador.

- Vamos a la página web de Moodle (http://moodle.org/) y seleccionamos el menú correspondiente en la zona de Downloads.

- Seleccionamos en nuestro caso "Moodle for Windows", y escogemos la versión más reciente de Moodle.

Windows Moodle Distribution

Version	Download
Moodle 1.9.5+ MOODLE_19_WEEKLY Built Weekly 6 days 9 hours ago	Download 65.8MB 1135 today

- El enlace nos proporciona el siguiente archivo comprimido, que debemos descomprimir para proceder a su instalación:



3.2.2. Instalación de Moodle

En este apartado describimos cómo instalar Moodle en nuestro ordenador de forma local.

- Hacemos clic en 'Start Moodle.exe' para iniciar la instalación:

  Dentro de la carpeta "MoodleWindowsInstaller-latest-19", buscamos el icono "Moodle".



Hacemos clic en el icono "Start Moodle", y esperamos a que la ventana que aparece en pantalla se cierre.

- Accedemos en nuestro explorador web a la dirección http://localhost/ para usar nuestro "sitio Moodle", visualizando la ventana que se muestra en la Figura 3.1.

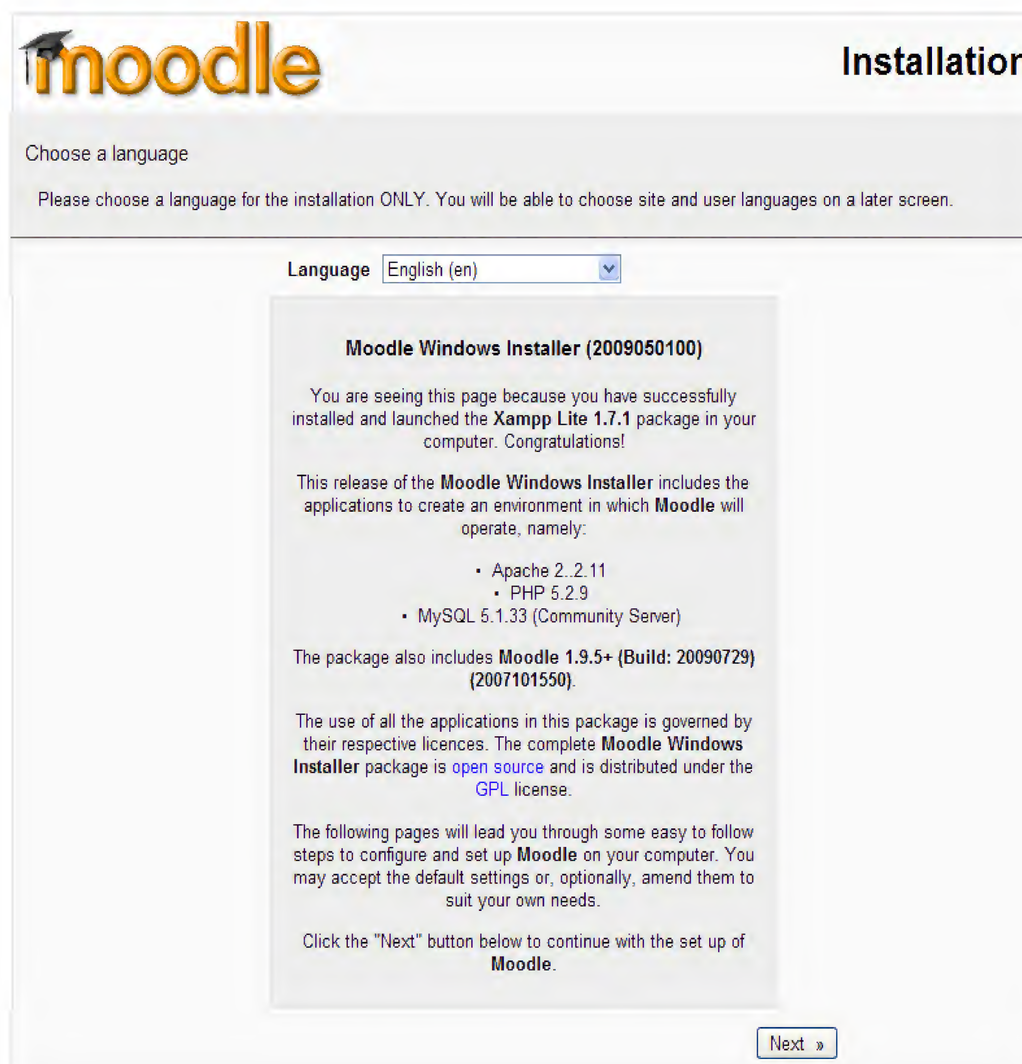
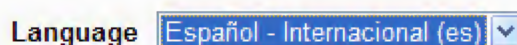


Figura 3.1. Instalación de Moodle.

En la página de inicio para instalar Moodle nos informan que utiliza el paquete Xampp Lite, el servidor Apache, la instalación del lenguaje de programación PHP y el gestor de bases de datos MySql. También nos informa que es un proyecto Open Source, con licencia GPL.

- Lo primero que hacemos es cambiar el idioma a nuestro "Moodle Site", seleccionado el español y pulsando el botón "Siguiente":



- Visualizaremos entonces la pantalla mostrada en la Figura 3.2.



Figura 3.2. Comprobaciones de Moodle.

Moodle mientras se va instalando nos va mostrando que todo el proceso va correcto. Terminadas estas comprobaciones hacemos click en el botón "Siguiente".

- Visualizamos la pantalla mostrada en la Figura 3.3.

Por favor, confirme las siguientes direcciones de la instalación de Moodle

Dirección Web: Especifique la dirección web completa en la que se accederá a Moodle. Si su sitio es accesible desde diferentes URLs entonces elija la más natural que sus estudiantes deberían utilizar. No incluya la barra final.

Directorio Moodle: Especifique la ruta completa de esta instalación. Asegurese de que las mayúsculas/minúsculas son correctas.

Directorio de Datos: Es necesario un lugar donde Moodle pueda guardar los archivos subidos. Este directorio debe ser legible y escribible por el usuario de su servidor web (normalmente 'nobody' o 'apache'), pero no debería ser directamente accesible desde la red.

Dirección Web http://localhost

Directorio Moodle C:\Documents and Settings\EMC1\Mis docum

Directorio de Datos C:\Documents and Settings\EMC1\Mis docum

« Anterior   Siguiente »

Figura 3.3. Configuración de dirección web y de datos.

Esta pantalla es fundamental para poder hacer que nuestro sitio Moodle instalado en nuestro PC pueda ser accedido desde el resto de Internet.

- Haciendo click en el botón "Siguiente, visualizamos la pantalla mostrada en la Figura 3.4.

Ahora necesita configurar los ajustes de la base de datos donde se almacenarán la mayoría de los datos de Moodle. El instalador creará la base de datos con los ajustes especificados más abajo.

Tipo: el valor por defecto es "mysql"
Servidor: el valor por defecto es "localhost"
Nombre: nombre de la base de datos, e.g., moodle
Usuario: el valor por defecto es "root"
Contraseña: contraseña de la base de datos
Prefijo de tablas: prefijo opcional para todas las tablas

Tipo mysql
Servidor localhost
Base de datos moodle
Usuario root
Contraseña
Prefijo de tablas mdl_

« Anterior   Siguiente »

Figura 3.4. Configurar base de datos de Moodle

Mediante esta figura configuramos la base de datos de Moodle introduciendo la contraseña y nombres correspondientes.

- Pulsando el botón "Siguiente", visualizaremos la pantalla mostrada en la Figura 3.5.

Comprobando su entorno

Estamos comprobando que diferentes componentes de su servidor cumplen con los requerimientos del sistema

Comprobaciones del Servidor

Nombre	Información	Informe	Estado
unicode		es obligatorio que esté instalado/activado	OK
database	mysql	versión 4.1.16 es obligatoria y está ejecutando 5.1.33	OK
php		versión 4.3.0 es obligatoria y está ejecutando 5.2.9	OK
php_extension	iconv	es recomendable que esté instalado/activado	OK
php_extension	mbstring	es recomendable que esté instalado/activado	OK
php_extension	curl	es recomendable que esté instalado/activado	OK
php_extension	openssl	es recomendable que esté instalado/activado	OK
php_extension	tokenizer	es recomendable que esté instalado/activado	OK
php_extension	xmlrpc	es recomendable que esté instalado/activado	OK
php_extension	ctype	es recomendable que esté instalado/activado	OK

« Anterior   Siguiente »

Figura 3.5. Comprobaciones del servidor durante la instalación de Moodle

- Seleccionando el botón "Siguiente", se nos muestra en pantalla correspondiente a la descarga del paquete de idioma seleccionado (Figura 3.6).

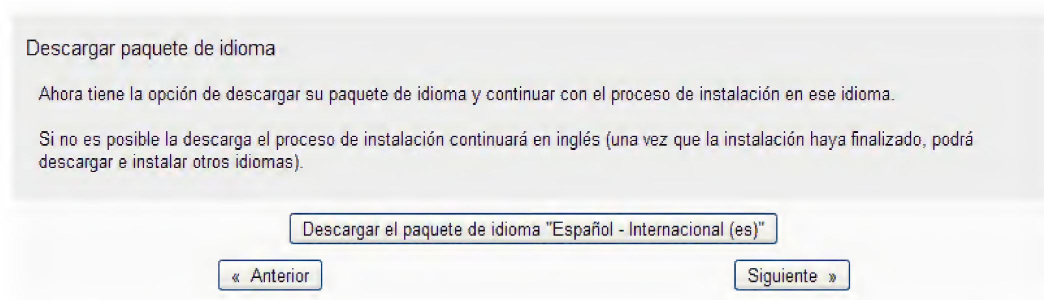


Figura 3.6. Descarga e instalación del paquete de idioma seleccionado

- Seleccionando el botón "Siguiente", Moodle nos informa que la configuración previa a la instalación ha finalizado correctamente (Figura 3.7).

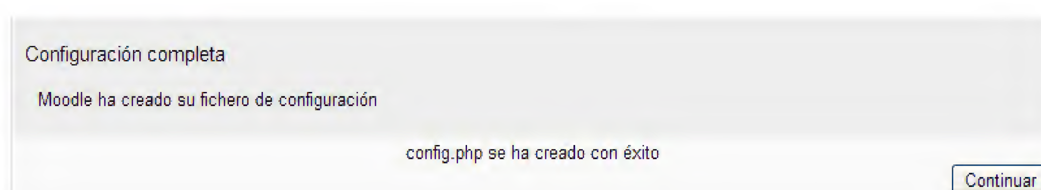


Figura 3.7. Finalización de la configuración de Moodle

- Pulsando el botón "Continue", visualizaremos la pantalla mostrada en la Figura 3.8, donde Moodle nos solicita la aceptación de las condiciones de instalación.

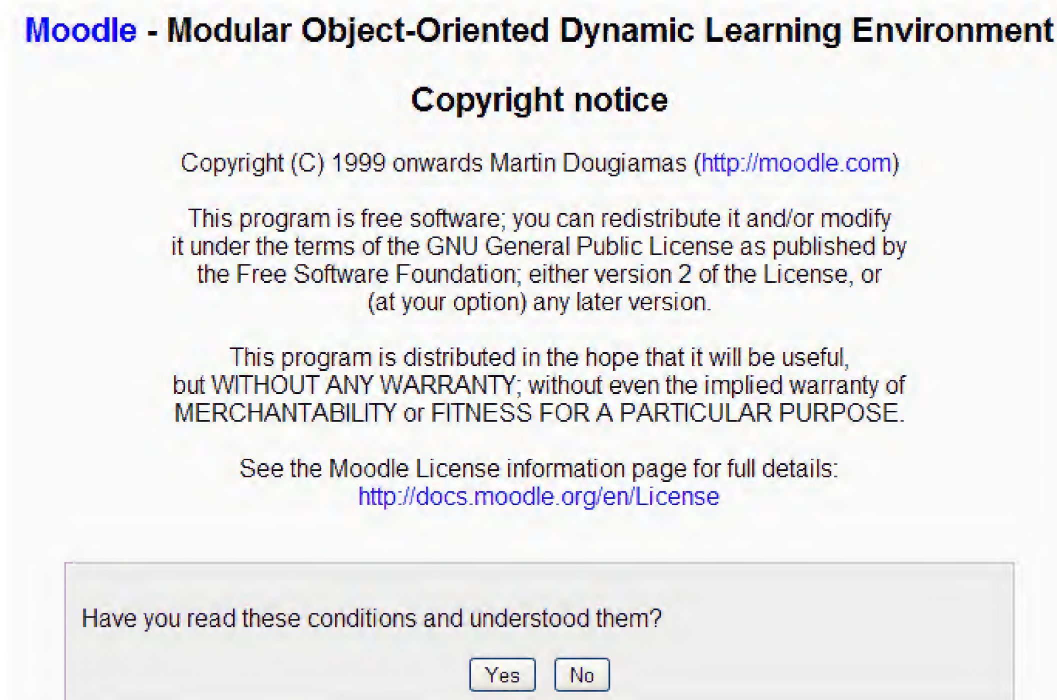


Figura 3.8. Aceptar condiciones de Moodle.

Tenemos que aceptar las condiciones para poder instalar el paquete Moodle.

- Tras pulsar el botón "Yes", Moodle nos informa sobre la versión que va a proceder a instalarse (Figura 3.9).

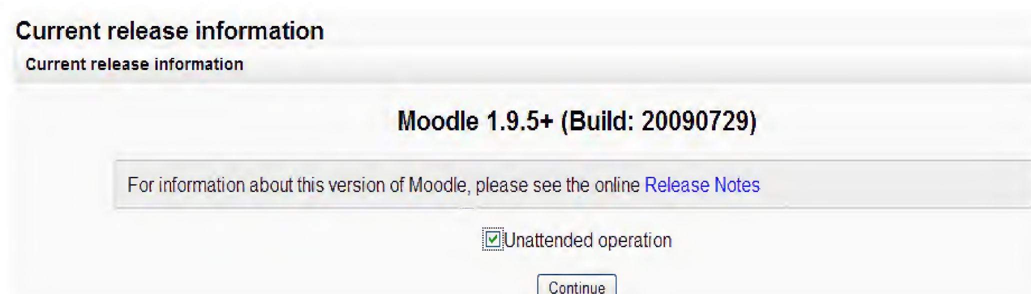


Figura 3.9. Información acerca de la versión de Moodle.

- Seleccionamos la opción *operation unattended* (operación desatendida) y pulsamos el botón "Continue", con lo que visualizaremos la pantalla mostrada en la Figura 3.10.

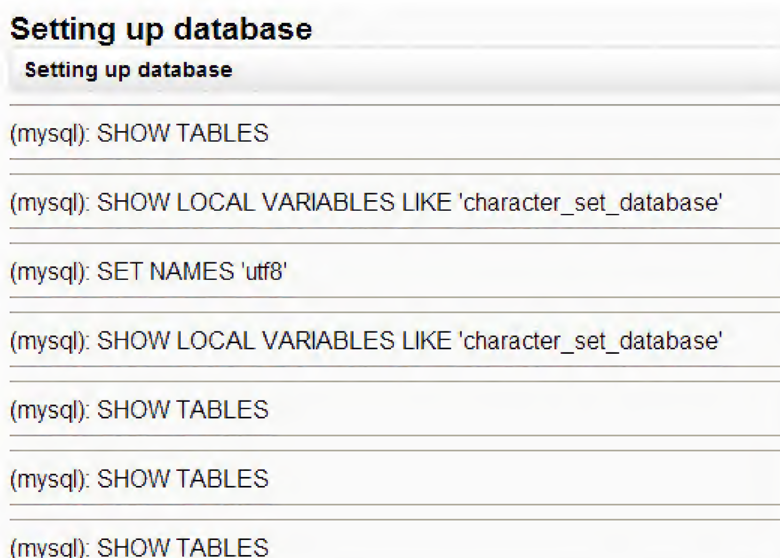


Figura 3.10. Configuraciones de instalación de Moodle.

Según vamos instalando Moodle nos va mostrando las tablas que se van creando. Pulsamos el botón "Continue" para todas las demás pantallas.

- Llegamos a la pantalla mostrada en la Figura 3.11, donde se nos solicita configurar los datos del administrador de Moodle :

Setup administrator account

On this page you should configure your main administrator account which will have complete control over the site. Make sure you give it a secure username and password as well as a valid email address. You can create more admin accounts later on.

General

* Show Advanced

Username* admin

New password* Unmask

Force password change

First name* Admin

Surname* User

Email address*

Email display Allow everyone to see my email address

Email activated This email address is enabled

City/town*

Select a country* Select a country...

Timezone Server's local time

Preferred language English (en)

Description

Trebuchet 1 (8 pt) Lang

Path:

Interests

List of interests

Picture of

Current picture None

Delete

New picture (Max size: 16MB) Examinar...

Picture description

Optional

* Show Advanced

Update profile

There are required fields in this form marked*.

Figura 3.11. Configuración de los datos de administrador de Moodle

- Introducimos la información requerida, tal y como muestra la Figura 3.12.

General

Username* admin

Choose an authentication method * Manual accounts

New password* 123456 Unmask

Force password change

First name* Eduardo

Surname* Rojo Sánchez

Email address* e100048464@gmail.com

Email display Allow everyone to see my email address

Email activated This email address is enabled

Email format* Pretty HTML format

Email digest type* No digest (single email per forum post)

Forum auto-subscribe* Yes: when I post, subscribe me to that forum

When editing text* Use HTML editor (some browsers only)

AJAX and Javascript* Yes: use advanced web features

Screen reader* No

City/town* Madrid

Select a country* Spain

Timezone Server's local time

Preferred language English (en)

Description

Trebuchet 3 (12 pt) Normal Lang

Desarrollandor Web.

Path: body » p

Picture of

Current picture None

Delete

New picture (Max size: 16MB) C:\Documents and Setting Examinar...

Picture description Foto de un patito.

Interests

List of interests XHTML, DHTML, JavaScripts, PHP, MySql, Ajax, Poser, DreamWeaver, Flash, Action Scripts,...

Figura 3.12. Introducción de la información del administrador en Moodle.

- Tras hacer clic en el botón "Update Profile", visualizaremos la página mostrada en la Figura 3.13, en la que podemos introducir información sobre nuestro sitio Moodle (Figura 3.14).

The settings shown below were added during your last Moodle upgrade. Make any changes necessary to the defaults and then click the "Save changes" button at the bottom of this page.

New settings - Front Page settings

Full site name
fullname

Short name for site (eg single word)
shortname

Front Page Description
summary

Path:

This description of the site will be displayed on the front page.

New settings - Manage authentication

Self registration
registerauth
Disable
Default: Disable

If an authentication plugin, such as email-based self-registration, is selected, then it enables potential users to register themselves and create accounts. This results in the possibility of spammers creating accounts in order to use forum posts, blog entries etc. for spam. To avoid this risk, self-registration should be disabled or limited by *Allowed email domains* setting.

Save Changes

Figura 3.13. Configuración de la información sobre nuestro sitio Moodle

Full site name
fullname
Universidad Carlos III de Madrid

Short name for site (eg single word)
shortname
UC3M

Front Page Description
summary

Sito Sloodle (Moodle + Sloodle) de la UC3M

Path: body » p

This description of the site will be displayed on the front page.

Figura 3.14. Introducción de la información sobre nuestro sitio Moodle

Introducimos el nombre de nuestro sitio Moodle y el nombre de nuestro sitio Moodle de forma corta.

- Hacemos click en el botón "Save Changes" y visualizando en nuestro explorador la web http://localhost/ podremos ver nuestro sitio Moodle (Figura 3.15).

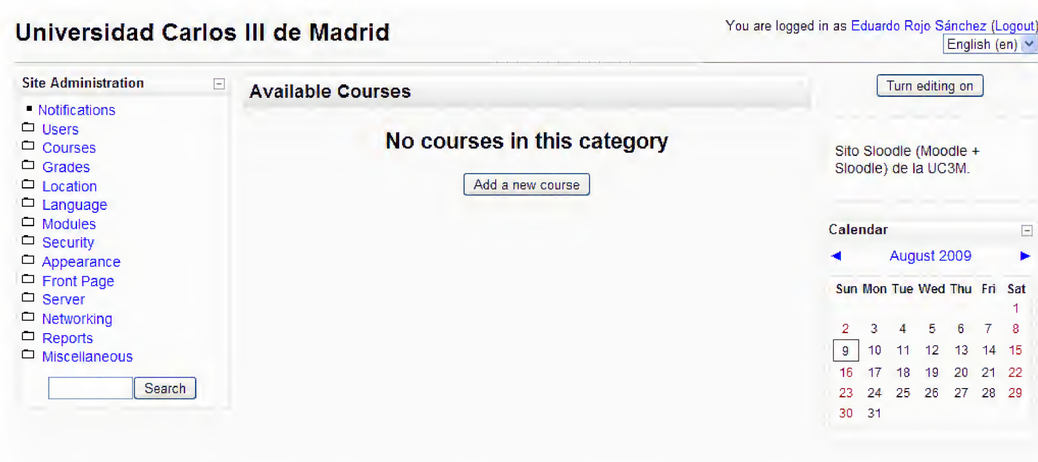


Figura 3.15. Pantalla inicial del sitio Moodle configurado

3.3. Sitios web gratuitos de prueba de Sloodle

3.3.1. Crear una cuenta de MoodleFarm

En este apartado explicamos cómo crear una cuenta gratuita en MoodleFarm, para conseguir así un sitio Web Sloodle.

- Accedemos a la dirección Web http://moodlefarm.socialminds.jp/, donde podremos visualizar la pantalla mostrada en la Figura 3.16.

Create a free demo Sloodle site now

Site name .moodlefarm.socialminds.jp

Your e-mail address

Fill with demo data? Yes No

Create free sloodle site

This demo is provided free of charge to help you try out Sloodle. It may be deleted at any time.

You can contact us here if you are looking for permanent hosting and support, or see the Sloodle website for advice on setting up the Sloodle module on your own server.

Figura 3.5. Creación de un sitio libre Sloodle en MoodleFarm

Cumplimentando los datos que nos solicitan, ya tendremos nuestro sitio web Sloodle. Cabe destacar que nos informan que la demostración se proporciona de forma gratuita para probar Sloodle y el servicio puede suprimirse en cualquier momento.

3.3.2. Creación de una cuenta en keytoschool

En este apartado vamos a configurarnos una cuenta en el servidor libre de Moodle keytoschool, que viene integrado con Sloodle. La principal ventaja de este servidor es que se ocupan periódicamente de actualizar tanto Moodle como Sloodle.

- Acudimos a la dirección web: http//www.keytoschool.com, donde podremos visualizar la pantalla mostrada en la Figura 3.17 para proceder al registro de una cuenta en este servidor.

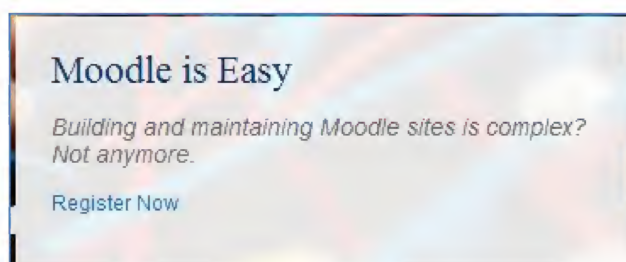


Figura 3.6. Página inicial del servidor Sloodle keytoschool

- Hacemos clic en el enlace **Register Now** para introducir el conjunto de datos solicitados (Figura 3.18).

Your Name * Your First & Last name	eduardo / rojo sanchez
Your E-Mail Address * A confirmation email will be sent to you at this address	100048464@alumos.uc3m.es
Choose a Login Name (User ID) * It must be 4 or more characters in length and may only contain small letters, numbers, and the underscore '_'	ifonsa
Choose a Password * Must be 4 or more characters	••••••••••
Confirm your password * Enter password again	••••••••••
Enter Verification Text * Please type text from image	gvc5 / gvc5
Country*	Spain
State*	Madrid
City*	madrid
Street*	fuentebella
ZIP*	28980
	Continue

Figura 3.7. Registro de una cuenta de usuario en el servidor Sloodle keytoschool

- Tras pulsar el botón “Continue”, se nos solicita introducir nuestro nombre de usuario y contraseña.

Enjoy your membership. To Login please click here Login

- Tras hacer clic en el enlace “Login”, visualizamos la pantalla mostrada en la Figura 3.19, donde seleccionamos la opción “Our sub-domain” como tipo de dominio.

Control Panel

Domain Type: *

Your own domain name

Your own sub-domain

Our sub-domain

If you are unsure, choose *Our sub-domain*. This can be changed later.

Moodle

Figura 3.8. Selección de tipo de dominio en el servidor keytoschool

- Tras pulsar el botón “Moodle”, seleccionamos un dominio disponible (Figura 3.20).

Control Panel

Available domains: *

edumoot.com

eduspoke.com

moodlehub.com

Sub Domain: *

yourschool

Sub-domain on which you want to install Moodle.

Moodle Go Back

The following page will take 60 seconds to load. Please wait after clicking *Moodle*

Figura 3.9. Selección de dominio en el servidor keytoschool

En esta pantalla introducimos el nombre del subdominio:

Sub Domain: *

UC3M

Sub-domain on which you want to install Moodle.

- Tras pulsar el botón Moodle, nos aparece el Panel de Control de la aplicación (Figura 3.21).

Control Panel

Full site name: *

Your School Name

Can be changed later in Moodle

Front Page Description: *

Add message to your users in a block on the front page of your site

Can be changed later in Moodle

My Moodle :)

Figura 3.10. Panel de control del servidor keytoschool

En esta página introducimos el nombre completo del sitio Moodle que vamos a crear y una breve descripción sobre él.

Full site name: *

Universidad Carlos III de Madrid

Can be changed later in Moodle

Front Page Description: *

Sitio dedicado a la investigación.

Can be changed later in Moodle

- Tras pulsar el botón "My Moodle :)", podremos visualizar la información actual de nuestra cuenta en el servidor (Figura 3.22).

Control Panel

FTP host	uc3m.edumoot.com
Creation Date	2010-04-07
Total Number of users	2
Total Number of courses	1
Total Size	1 MB
Total Bandwidth	1 MB
FTP username	xvlhikd6

Advanced options (see below) should not be used until you are sure what they do. You can discuss about them in **forums**.

▸ Advanced Options

Figura 3.11. Información sobre la cuenta creada en el servidor keytoschool

<u>Nota</u>: “Total número de usuarios” significa que disponemos de 2 usuarios por defecto, y 1 MB de “tamaño total”, indica el tamaño de lo que tenemos por ahora almacenado.

- <u>Tras hacer clic en el enlace uc3m.edumoot.com, podremos visualizar nuestro sitio Moodle (Figura 3.23).</u>

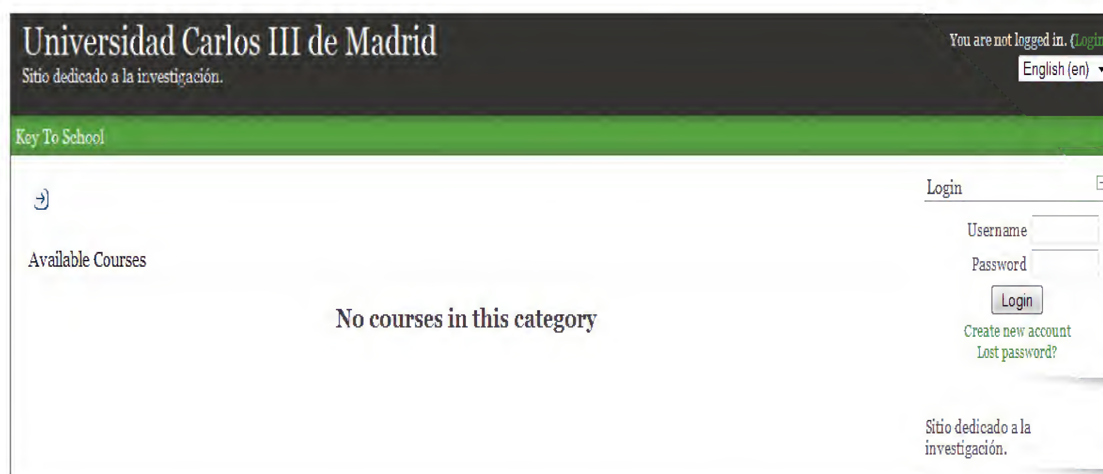


Figura 3.12. Página inicial de nuestro servidor Moodle en keytoschool

- <u>Tras introducir nuestro usuario y contraseña en la sección de login, se nos informa sobre la configuración de nuestro servidor (Figura 3.24).</u>

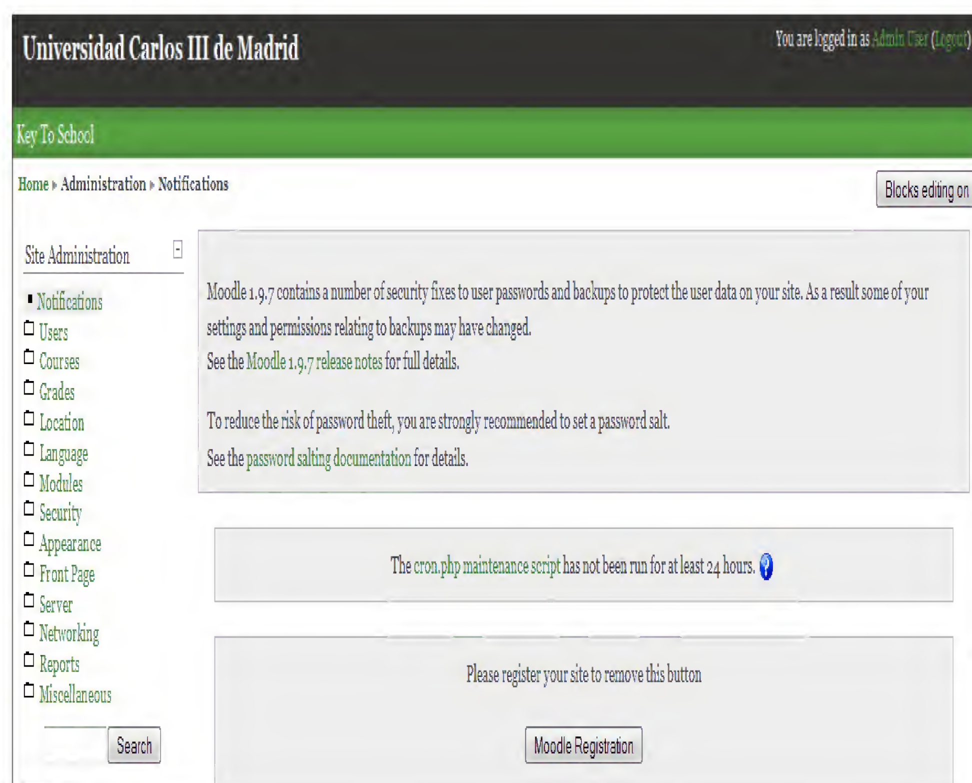


Figura 3.13. Configuración de nuestro servidor Moodle en keytoschool

Nuestro sitio Web de Moodle ya está configurado, y se nos solicita que lo registremos. También se nos informa que el fichero *cron.php* no ha sido ejecutado en al menos 24 horas.

3.4. Instalación de CRON

Algunos módulos de Moodle requieren revisiones continuas para llevar a cabo determinadas tareas. Por ejemplo, Moodle necesita revisar los foros de discusión para poder enviar copias de los mensajes a los usuarios que están suscritos. El script que se encarga de estas tareas está disponible en el directorio *admin*, y se llama *cron.php*. De este modo, es importante que este archivo se ejecute regularmente. La carga de este script no es muy alta, así que un intervalo de cinco minutos es razonable generalmente.

La forma más sencilla de ejecutar este fichero consiste en la utilización del paquete *moodle-cron-for-windows.zip*, que hace que esta tarea se pueda llevar a cabo fácilmente mediante la instalación un servicio Windows.

A continuación describimos los pasos a realizar para proceder a la instalación de Cron.

- <u>Accedemos como administrador a nuestro sitio web y comprobamos las notificaciones</u>:

Site Administration

▪ Notifications

En este apartado veremos la notificación que nos indica que el script del *cron.php* no ha sido ejecutado en al menos 24 horas.

The cron.php maintenance script has not been run for at least 24 hours.

- Comprobamos que el script funciona en el navegador para nuestro sitio web, accediendo para ello a la siguiente URL:

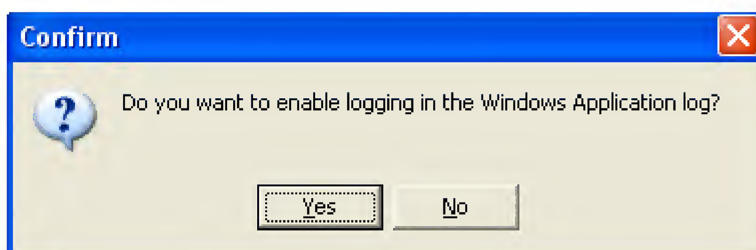
  http://10004846.moodlefarm.socialminds.jp/admin/cron.php

Como resultado de esta comprobación, el navegador web deberá mostrar por pantalla el siguiente mensaje:

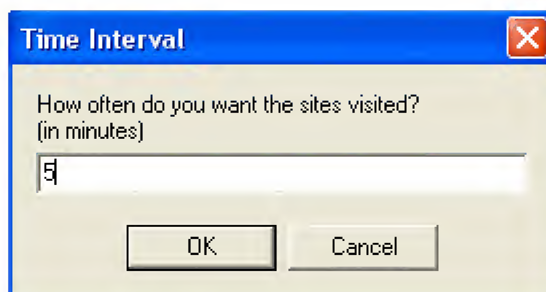
```
Cron script completed correctly
```

- Procedemos a la instalación del programa *MoodleCron-Setup.exe*, utilizando para ello el ejecutable suministrado.

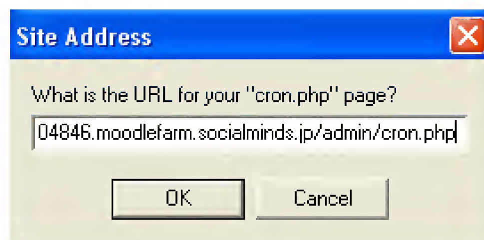
Como primer paso, confirmamos que deseamos utilizar la funcionalidad de almacenamiento en archivos log:



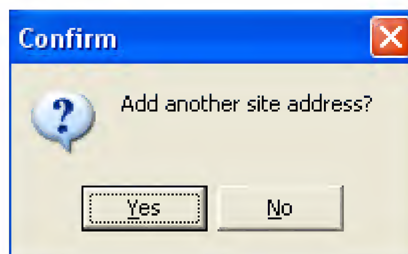
Seguidamente, introducimos cinco minutos como tiempo entre ejecuciones sucesivas del script:



A continuación indicamos la dirección donde se encuentra el fichero cron.php en nuestra página web:



Por último, indicamos que no queremos gestionar ningún otro cron de otra página web. Con ello, tendremos instalada satisfactoriamente esta utilidad.



3.5. Creación de una cuenta de correo para la autentificación de nuestro servidor

La confirmación por correo electrónico es el método de autenticación predeterminado. Cuando el usuario se inscribe en los servidores descritos en el apartado anterior, escogiendo su propio nombre de usuario y contraseña, se le remite automáticamente un email de confirmación a su dirección de correo electrónico.

Este email contiene un enlace seguro a una página donde el usuario puede confirmar su cuenta. En las siguientes entradas se comprueban el nombre de usuario y contraseña contra los valores guardados en la base de datos de Moodle.

3.5.1. Creación de una cuenta de correo electrónico de Yahoo

En este apartado describimos cómo crear una cuenta de correo en el servidor Yahoo y habilitar el servicio POP.

- Accedemos a la página web http://es.yahoo.com/ y seleccionamos el enlace Correo .
- En el menú de cuenta de correo Yahoo hacemos clic en opciones y en opciones de correo (Opciones / Opciones de correo).
- Seleccionamos "Acceso POP y reenvío", y dentro de este menú, la opción "Configurar o modificar el Acceso POP y reenvío:

Acceso POP y reenvío

Configurar o modificar el Acceso POP y reenvío

- Con ello, veremos la pantalla de configuración de acceso POP, que deberemos configurar tal y como muestra la Figura 3.25.

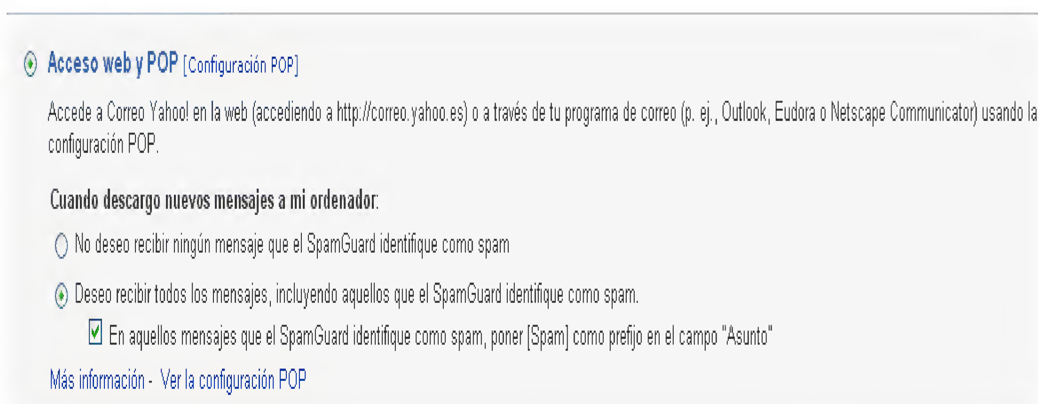


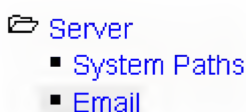
Figura 3.14. Configuración de una cuenta de correo Yahoo

- En la cuenta de Yahoo nos proporcionan los siguientes datos:
  - **Servidores SMTP:** smtp.correo.yahoo.es
  - **Nombre de usuario SMTP:** eduardorojosanchez@yahoo.es
  - **Contraseña SMTP:** ********
  - **Email de apoyo:** eduardorojosanchez@yahoo.es

3.5.2. Configuración del uso de la autentificación e-mail en nuestro sitio Moodle

Para configurar el uso de la autentificación e-mail en nuestro sitio Moodle cabe seguir los siguientes pasos.

- Accedemos como administrador a nuestro sitio Web y seleccionamos Server/Email:



- En la página Web que se nos muestra, introducimos los datos de la cuenta Yahoo creada en el apartado anterior:

SMTP hosts	smtp.correo.yahoo.es
SMTP username	eduardorojosanchez@yahoo.es
SMTP password	☑ Unmask
Support email	eduardorojosanchez@yahoo.es

- Finalizamos la sesión de Administrador en nuestro sitio Web:

Admin User (Logout)

3.5.3. Creación de una cuenta de usuario en nuestro sitio web Moodle

En este apartado vamos a crear una cuenta de usuario en nuestro sitio web Moodle.

- Pulsamos el botón "Comience ahora creando una cuenta":

Registrarse como usuario

Comience ahora creando una cuenta

Con ello, visualizaremos la página mostrada en la Figura 3.26.

Crear un nuevo usuario y contraseña para acceder al sistema

Nombre de usuario* giaa

Contraseña* ☑ Desenmascarar

Por favor, escriba algunos datos sobre usted

Dirección de correo* e100048464@hotmail.com

Correo (de nuevo)* e100048464@hotmail.com

Nombre* Eduardo

Apellido* Rojo Sánchez

Ciudad* Madrid

País* España

Crear cuenta | Cancelar

Figura 3.15. Registro de una cuenta de usuario en nuestro sitio Moodle

- Tras pulsar el botón "Crear Cuenta", se nos mostrará por pantalla la siguiente información:

Hemos enviado un correo electrónico a **e100048464@hotmail.com**

En él encontrará instrucciones sencillas para concluir el proceso.

Si tuviera alguna dificultad, contacte con el Administrador del Sistema.

- Accedemos al webmail de Yahoo para consultar el correo remitido:



Confirmación de la cuenta xmpl

De: Admin User (eduardorojosanchez@yahoo.es)
Es posible que no conozcas a este remitente. Marcar como seguro | Marcar como correo no deseado
Enviado: viernes, 31 de julio de 2009 10:52:33
Para: Eduardo Rojo Sánchez (e100048464@hotmail.com)

Hola, Eduardo Rojo Sánchez.

Se ha solicitado la apertura de una cuenta en 'xmpl' utilizando su dirección de correo.

Para completar el proceso de inscripción haga clic aquí:

http://10004846.moodlefarm.socialminds.jp/login/confirm.php?data=EYzhkKnB7RYkGQf/giaa

En la mayoría de programas de correo electrónico este enlace debería aparecer en azul. Si no funciona, córtelo y péguelo en la ventana de direcciones de su navegador.

Si necesita ayuda, contacte por favor con el administrador del sitio,

Admin User
eduardorojosanchez@yahoo.es

Figura 3.16. Correo remitido para el registro de nuevos usuarios en nuestro sitio Moodle

- <u>Finalizamos el registro del nuevo usuario tras hacer clic en la dirección web que se nos indica.</u>

Gracias, Eduardo Rojo Sánchez

Registro confirmado

Cursos

3.6. Creación y Configuración de un curso

3.6.1. Creación de un curso

En este apartado detallamos cómo crear un curso en nuestro servidor Moodle.

- <u>Accedemos como Administrador a nuestro sitio Web Moodle y seleccionamos el menú “Courses / Add/edit courses ”</u>:

Courses
- Add/edit courses

- <u>Creamos una categoría nueva, pulsando el botón “Add new category”, completando para ello los datos que se nos solicitan en la Figura 3.28.</u>

Figura 3.17. Creación de una nueva categoría de cursos en nuestro servidor Moodle

- Tras ello, podemos comprobar cómo se ha creado la nueva categoría "Cursos Libres" (Figura 3.29).

Figura 3.18. Categorías de cursos disponibles en nuestro servidor Moodle

- Hacemos clic en la nueva categoría que hemos creado ("Cursos Libres").

- Seleccionamos la opción “Add a new course” y editamos la información sobe el curso que vamos a crear (Figura 3.30).

Figura 3.19. Edición de la información general del curso Moodle

- Seleccionamos el Formato por temas:

- Indicamos que el número de temas a mostrar van a ser 6:

- Cualquier invitado podrá acceder sin utilizar clave:

- Seleccionamos el lenguaje Español – Internacional:

- A continuación debemos asignar roles a nuestro curso (Figura 3.31).

Asignar roles en Curso: OpenSim y OsGrid

Roles	Descripción	Usuarios
Administrator	Administrators can usually do anything on the site, in all courses.	0
Course creator	Course creators can create new courses and teach in them.	0
Teacher	Teachers can do anything within a course, including changing the activities and grading students.	0
Non-editing teacher	Non-editing teachers can teach in courses and grade students, but may not alter activities.	0
Student	Students generally have fewer privileges within a course.	0
Guest	Guests have minimal privileges and usually can not enter text anywhere.	0

Figura 3.20. Asignación de roles en el curso Moodle

- Vamos a asignar a un profesor en nuestro curso, seleccionamos para ello el enlace "Course creator" (Figura 3.32):

Asignar roles en Curso: OpenSim y OsGrid

Rol a asignar: Course creator

Período de vigencia de la matrícula: Sin límite — Comienzo en: Hoy (31 de julio de 2009)

0 usuarios existentes — 5 usuarios potenciales

Eduardo Rojo Sánchez, e100048464@hotmail.com
Student1 Student1, student1@example.com
Teacher1 Teacher1, teacher1@example.com
Admin User, online@edochan.com
Admin User, e100048464@gmail.com

◄ Agregar — Quitar ►

Figura 3.21. Creación de un usuario Profesor en el curso Moodle

- Seleccionamos el usuario *Eduardo Rojo Sánchez* y hacemos clic en "Agregar", tal y como muestra la Figura 33.

Eduardo Rojo Sánchez, e100048464@hotmail.com

◄ Agregar — Quitar ►

Student1 Student1, student1@example.com
Teacher1 Teacher1, teacher1@example.com
Admin User, online@edochan.com
Admin User, e100048464@gmail.com

Figura 22. Selección del usuario Profesor en el curso Moodle

- Pulsando el botón OS visualizamos la pantalla inicial del curso creado.

Tal y como muestra la Figura 3.34, podemos ver que el curso contiene los tres temas que se decidió incluir. Por defecto, contiene además el foro "Novedades".

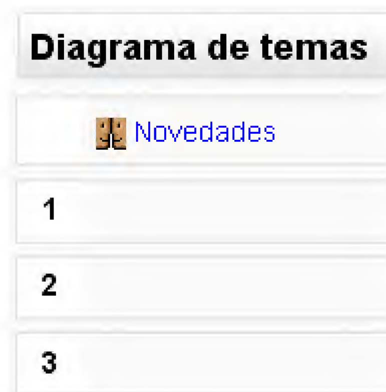
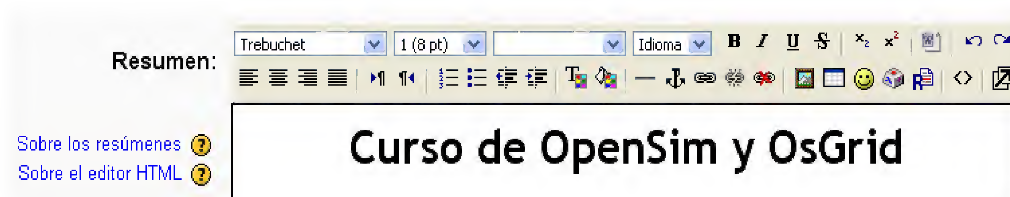


Figura 3.23. Aspecto inicial del curso Moodle creado

- Vamos a editarlo para cambiarlo de nombre, para ello hacemos clic en el botón "Activar edición".

Seguidamente seleccionamos el icono de edición (representado por una mano con un lápiz) e introducimos el texto "Curso de OpenSim y OsGrid".



- En el siguiente cuadro hacemos clic de nuevo sobre el icono de edición e introducimos el nombre del foro del curso.



Nombre del foro* Foro Principal de OS y OG

3.6.2. Creación de los temas de un curso

3.6.2.1. Creación de un índice de un curso

Vamos a crear el índice de nuestro curso.

- En el Tema 0 elegimos la opción "Editar una página Web":

Editar una página web

- Nos aparece la ventana mostrada en la Figura 3.35, en la que introducimos como **Nombre** índice, y escribimos un **Resumen** de esta página web:

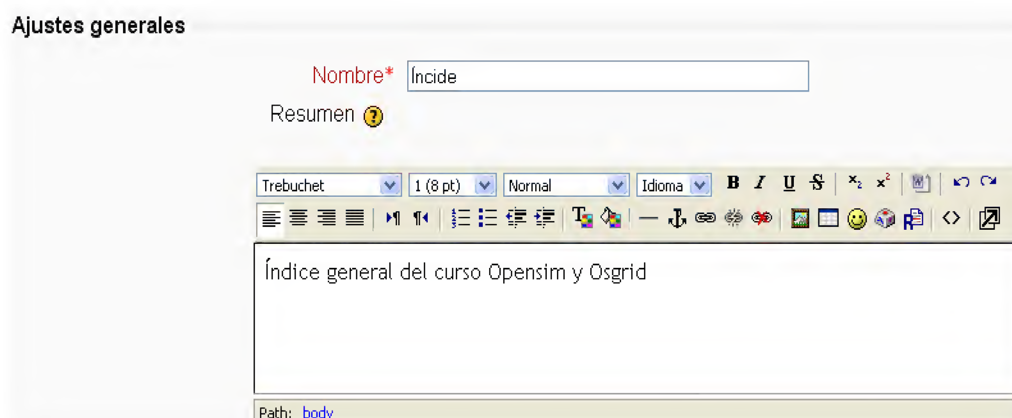


Figura 3.24. Ajustes generales del índice del curso

- Seleccionamos la sección "Editar una página de texto", y escribimos el índice de nuestro curso Moodle (Figura 3.36).

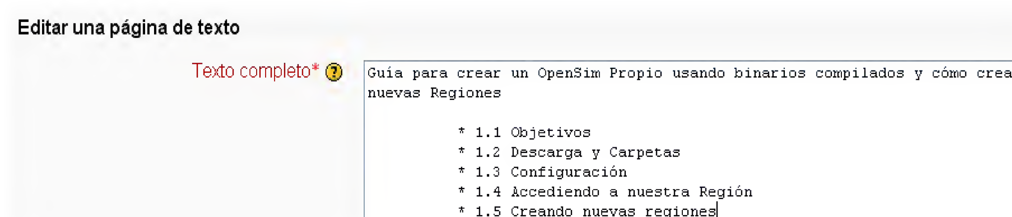


Figura 3.25. Índice de nuestro curso Moodle

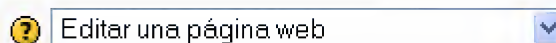
- Elegimos como Formato el de "Formato de texto plano":



3.6.2.2. Creación una página web

Vamos a crear en este apartado una página web donde se muestren todos los enlaces a las diferentes páginas web del curso.

- Seleccionamos la opción "Editar una página web":



- Al igual que en el apartado anterior, visualizaremos una pantalla en la que debemos introducir el Nombre y Resumen de esta página web (Figura 37).

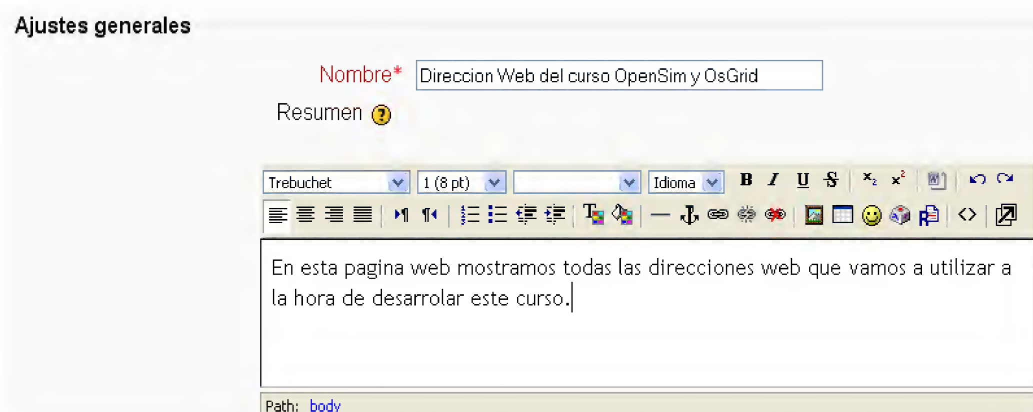


Figura 3.26. Ajustes generales de la nueva página web del curso

- Utilizamos el editor para introducir el código HTML de nuestra página (Figura 3.38).

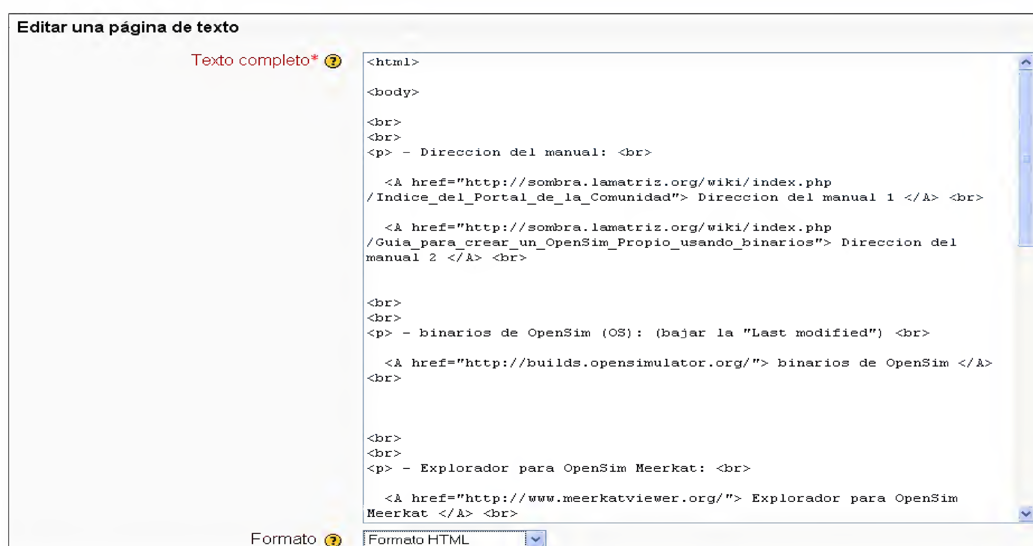


Figura 3.27. Edición del código HTML de la nueva página web del curso

- Introducimos las opciones de visualización de la nueva página. Para ello, en el menú "Ventana" seleccionamos "Nueva ventana" y hacemos clic sobre el enlace "Mostrar avanzadas" (Figura 3.39).

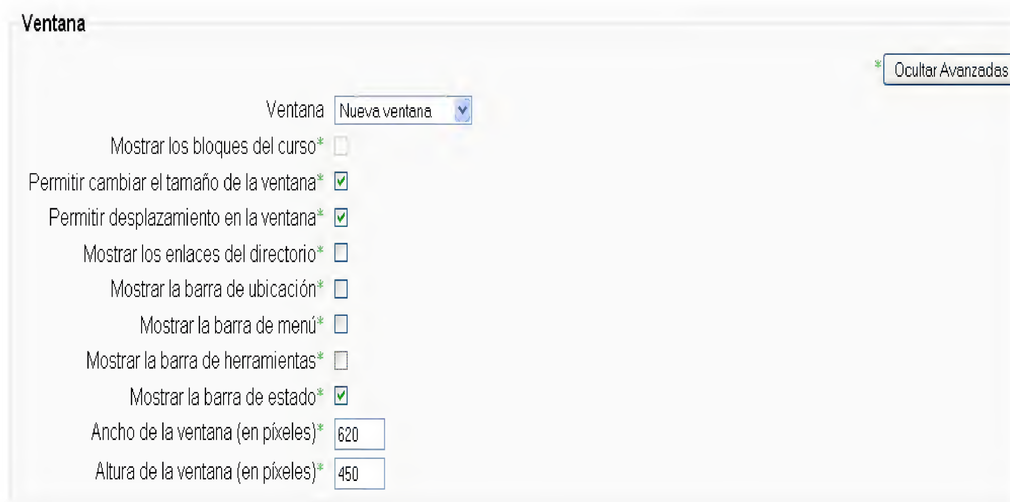
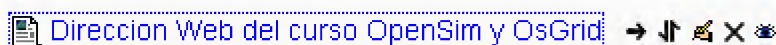


Figura 3.28. Edición del código HTML de la nueva página web del curso

Puede observarse que hemos deseleccionado algunas opciones en los "check box" de la ventana web por razones de seguridad.

- Seleccionamos el enlace web "Dirección Web del curso OpenSim y OsGrid"



- Con ello, visualizaremos la página web de Moodle en la que se muestra un índice de los contenidos de nuestro curso (Figura 3.40).

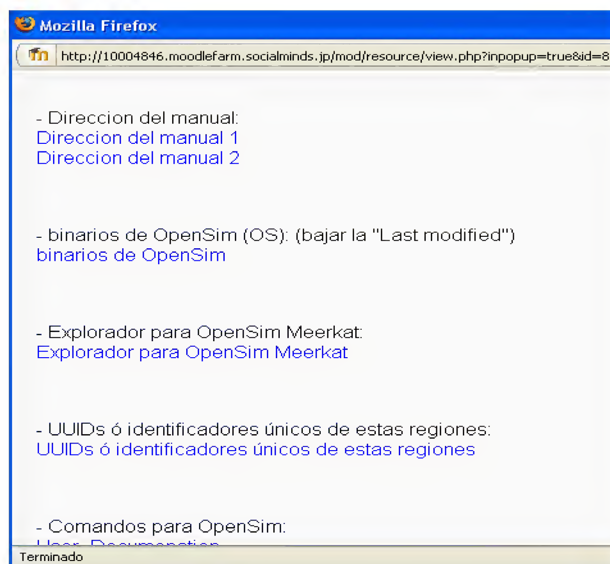
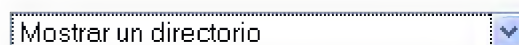


Figura 3.29. Página web con el índice de contenidos de nuestro curso Moodle

3.6.2.3. Edición del directorio principal

Vamos a editar el directorio principal de nuestro curso.

- Seleccionamos el enlace "Mostrar un directorio".



- Introducimos el Nombre *Directorio y Archivos*, y seleccionamos la opción "Ocultar".

Nombre* Directorio y Archivos

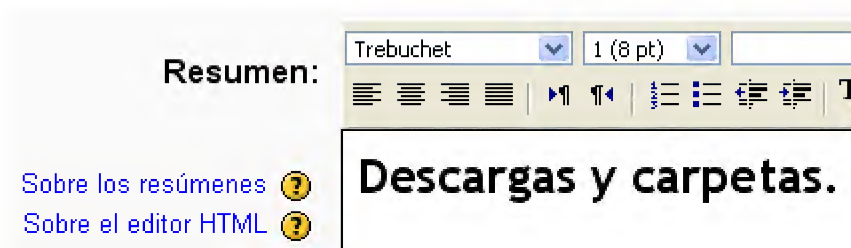
Visible Ocultar

- Con ello, dispondremos del directorio en nuestro curso.



3.6.2.4. Asignación de un nombre a un tema del curso

Para asignar un nombre a un tema de un curso, seleccionamos por el ejemplo el icono de edición (mano con lápiz) del Tema 1 y escribimos en el resumen *Descargas y carpetas*.



3.6.2.5. Almacenamiento de ficheros y creación de directorios

Vamos a crear un directorio para almacenar diversos tipos de documentos.

- Creamos un nuevo archivo, al que llamamos "*Apuntes de Descargas y carpetas*":

Nombre* Apuntes de Descargas y carpetas

- Seguidamente lo mostramos en el directorio 1, el cual depende del directorio principal. De esta forma, todas nuestras carpetas dependerán del directorio principal.

Mostrar un directorio 1

- Hacemos clic en el enlace "*Apuntes de Descargas y carpetas*".



- Pulsamos el botón *Editar archivos*.

Editar archivos

- Pulsamos el botón *Subir un archivo*.

Subir un archivo

- En la pantalla que se nos muestra, seleccionamos el archivo (en este caso, con nombre *Descarga y Carpetas.doc) y lo subimos a nuestro servidor (Figura 41).*

Subir un archivo (Tamaño máximo: 2Mb) --> /

Examinar...

Subir este archivo

Cancelar

Archivo subido con éxito

Nombre	Tamaño	Modificado	Acción
1._Descarga_y_Carpetas.doc	84.5Kb	31 jul 2009, 02:25	Renombrar

Con los archivos escogidos...

Crear un directorio | Seleccionar todos | No seleccionar ninguno | Subir un archivo

Figura 3.30. Almacenamiento de ficheros en el servidor de nuestro curso

- Seleccionamos la opción *Crear un directorio*.

Crear un directorio

- Simplemente debemos introducir su nombre.

Crear un directorio en /:

1 | Crear | Cancelar

Nombre

 1

3.6.2.6. Agregar un chat a nuestro curso

Cada tema del curso tiene su Chat propio.

- Seleccionamos la opción Chat.

Chat

- Introducimos en el Nombre de la sala "*Chat Tema 1. Descargas y carpetas*" y en el Texto introductorio "*Chat sobre Descargas y carpetas*" (Figura 3.42).

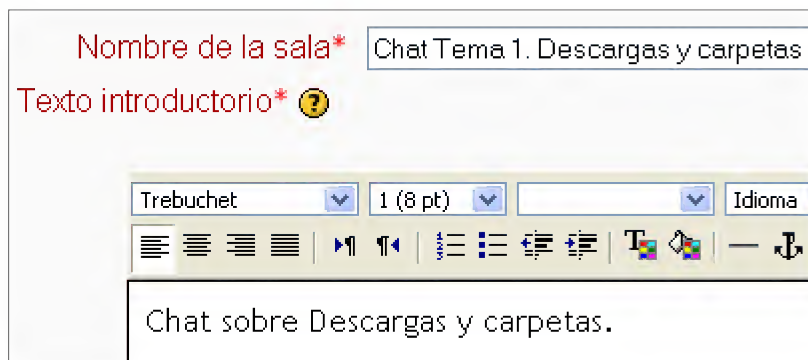
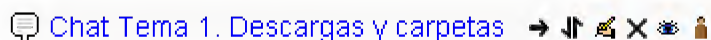


Figura 3.31. Configuración del chat de un tema del curso

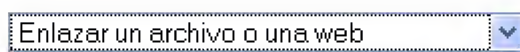
- Con ello, visualizraemos el chat creado para el tema.



3.6.2.7. Creación de un enlace a una página web

Mostramos ahora cómo crear un enlace a una página web.

- Seleccionamos en el tema 1:



- Asignamos un nombre para hacer referencia a la página web que queremos mostrar:

Nombre* Descargar binarios de OpenSim

- Introducimos la dirección web donde se encuentra la nueva página:

Ubicación http://builds.opensimulator.org/

- Ya hemos conseguido con ello el enlace a la página web:



3.7. Cuestionarios

3.7.1. Creación de un cuestionario

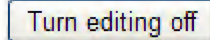
Un cuestionario de un curso Moodle puede definirse como una página web a modo de contenedor de preguntas de varios tipos extraídas del banco de preguntas y colocadas en un cierto orden.

Los cuestionarios de Moodle tienen dos componentes principales:

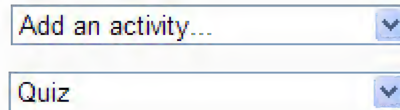
- ➢ El cuerpo del cuestionario (con todas sus opciones).
- ➢ Las preguntas que los estudiantes deben responder.

Mostramos el proceso para crear el cuerpo de un cuestionario:

- Activamos primero de todo el modo de edición.



- Seleccionamos la opción *"Cuestionario"* en la lista desplegable *"Agregar una actividad"*.

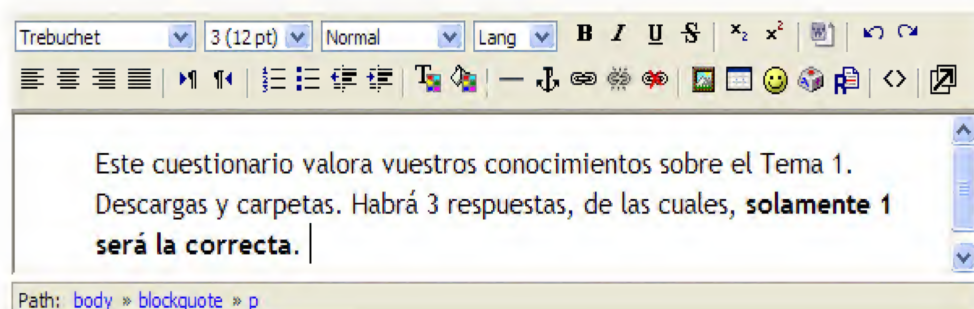


- Con ello, accederemos a la página de configuración de nuestro formulario:

  - **Nombre**: texto identificativo con el que quedará enlazado el cuestionario en la página principal del curso, dentro de la sección elegida. Se pueden incluir marcas o etiquetas HTML para resaltar parte o la totalidad del nombre.

Name* Quiz Tema 1. Descargas y carpetas.

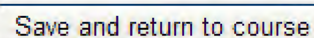
  - **Introducción**: texto que se mostrará a los estudiantes cada vez que inicien el cuestionario. Se puede utilizar para describir la finalidad del cuestionario, incluir instrucciones de resolución, indicar los criterios de calificación o cualquier otra cuestión metodológica.



  - **Número máximo de preguntas por página**: en cuestionarios extensos puede ser práctico limitar el número de preguntas por página. Cuando se añaden preguntas al cuestionario se insertarán automáticamente saltos de página de acuerdo con el ajuste que se introduzca aquí. También existe la posibilidad de cambiar los saltos de página más tarde en la página de edición.

Questions per page 10

- Pulsamos el botón *"Save and return to course"*



- Con ello, vemos el nuevo "Quiz (cuestionario)" que hemos agregado al curso.



3.7.2. Generación de una lista de preguntas

Describimos a continuación cómo generar una lista de preguntas.

- Hacemos clic sobre el enlace "*Quiz Tema 1. Descargas y carpetas*":

Quiz Tema 1. Descargas y carpetas.

- Se presentará en pantalla la página de edición del cuestionario, que mostramos en la Figura 3.43.

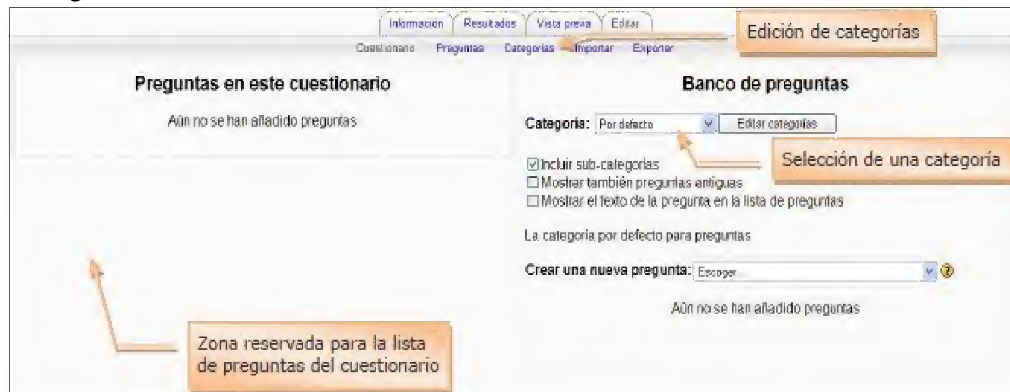


Figura 3.32. Edición de un Formulario en Moodle

- En esta página encontramos dos secciones:
  - A la izquierda se muestra un listado con las preguntas del cuestionario (inicialmente esta lista estará vacía).
  - A la derecha se encuentra el banco o repositorio de preguntas, organizado por categorías. Desde este banco se pueden copiar preguntas que pueden utilizarse en cualquier cuestionario, independientemente del curso.

- La pantalla de edición de preguntas muestra las preguntas de la categoría seleccionada actualmente. Podemos escoger una categoría en el menú desplegable *Categoría*.

Category Default for OpenSim y OsGrid

- Mediante la activación de la casilla de verificación "*Incluir sub-categorías*", Moodle mostrará también las preguntas de todas las subcategorías.

☐ Display questions from sub-categories too

- Puede observarse que al principio de cada curso sólo existe una categoría denominada *Por defecto…*

Category Default for OpenSim y OsGrid

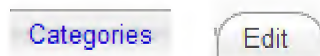
Es una buena costumbre crear categorías para organizar con detalle las preguntas y evitar mantener así que todas ellas aparezcan en una única lista sin clasificar. También

podemos implementar una jerarquía de categorías, creando para ello todas las subcategorías que se requieran dentro de una categoría padre.

3.7.2.1. Creación y edición de categorías

Vamos a crear categorías para las preguntas de nuestros cuestionarios.

- Para añadir o editar categorías hay que pulsar en el enlace *Categorías* o en el botón **Editar categorías**:



- Con ello, veremos la página *Editar categorías* (Figura 3.44).

Info Results Preview Edit
Quiz Questions Categories Import Export

Edit categories

Question Categories for 'Activity module: Quiz Tema 1. Descargas y carpetas.'

- Default for Quiz Tema 1. Descargas y carpetas. (0) The default category for questions shared in context 'Quiz Tema 1. Descargas y carpetas.'.

Question Categories for 'Course: OS_1'

- Default for OpenSim y OsGrid (0) The default category for questions shared in context 'OpenSim y OsGrid'.

Question Categories for 'Category: Miscellaneous'

- Default for Miscellaneous (0) The default category for questions shared in context 'Miscellaneous'.

Question Categories for 'System'

- Default for System (0) The default category for questions shared in context 'System'.

Add category

Parent: Default for OpenSim y OsGrid
Name*
Category info

Add category

There are required fields in this form marked*.

Figura 3.33. Edición de las categorías de las preguntas de los formularios

- En la parte inferior de esta página aparece la opción "*Add category*" y debajo varios campos vacíos que habrá que rellenar para crear la nueva categoría:

- **Parent**: permite situar la categoría como una subcategoría de otra. Si no hay más, ésta se colocará directamente en la parte superior.

Seleccionamos que la categoría que estamos añadiendo tenga como "*parent*" (padre), la categoría "*Default for Quiz Tema 1. Descargas y carpetas*".

Parent ? Default for Quiz Tema 1. Descargas y carpetas.

- **Name**: nombre de la categoría.

Name* Categoria 1ª del tema 1.

- **Category info**: breve descripción de la categoría.

Category info Preguntas fáciles del Tema 1.

- Tras pulsar el botón "*Add Category*", la nueva categoría quedará guardada y aparecerá en el listado de categorías (Figura 3.45).

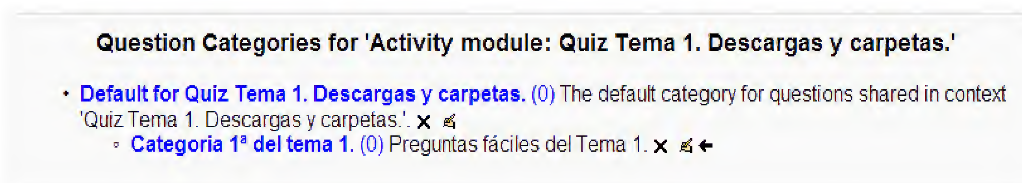


Figura 3.34. Listado de las categorías de las preguntas de los formularios

Este listado muestra las categorías existentes con sus respectivos atributos, que pueden editarse haciendo clic en su nombre. Desde esta página también se puede borrar una categoría existente mediante el icono de *Borrado* ✕. Si esta categoría contiene preguntas, las preguntas no se borrarán, sólo la categoría. El sistema preguntará a qué categoría se moverán esas preguntas y, a continuación, ejecutará la acción de borrar.

3.7.2.2. Creación y edición de preguntas

Una vez creadas las categorías, es el momento de añadir las preguntas.



- Para crear nuevas preguntas dentro de una categoría, seleccionaremos la categoría de la lista desplegable *Categoría (category)*.

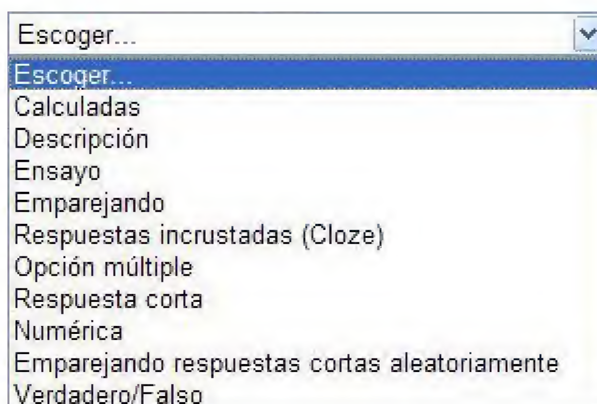
Seleccionamos de la lista desplegable la categoría "Categoría 1ª del tema 1".

Category Categoria 1ª del tema 1.

- <u>A continuación, escogemos un tipo de pregunta de la lista de selección "*Crear una nueva pregunta (Create new question)"*</u>:

Create new question Choose...

➢ Los tipos de preguntas posibles son:



- <u>Nosotros seleccionamos "*Opción Múltiple"* (Multiple Choice)"</u>. Se trata de preguntas en las que se da a elegir entre varias respuestas, pudiendo ser válidas más de una.

Create new question Multiple Choice

Después de elegir el tipo de pregunta deseado, visualizaremos el correspondiente formulario de configuración, que es específico para cada tipo de pregunta. Todas las preguntas tienen una serie de características comunes:

➢ Cada pregunta individual se identifica mediante un nombre (que debe ser único).

➢ Los enunciados de las preguntas (y de las respuestas) pueden incluir distintos formatos de caracteres, imágenes, tablas…, es decir, todo lo que permita el editor de texto HTML integrado en Moodle. También se pueden usar expresiones matemáticas o incrustar archivos multimedia (audio, vídeo, animaciones, etc.).

➢ Se puede incluir una imagen extra, externa, asociada a cada pregunta.

➢ Se puede asociar a cada respuesta un texto de refuerzo, pista o explicación. Este texto se muestra al estudiante cuando revisa la pregunta o el cuestionario y debería servir para orientar al alumnado en futuros intentos.

Además de estas opciones comunes, cada tipo de pregunta contiene tendrá una serie de características específicas. En las preguntas de opciones múltiples, el

estudiante debe elegir una o varias respuestas. Moodle ofrece una gran flexibilidad a la hora de crear este tipo de pregunta. De hecho, podemos crear preguntas de respuesta única y múltiple, presentar imágenes en la pregunta, ponderar las respuestas individuales, etc.

Hay dos tipos de preguntas de opción múltiple:

- de respuesta única.
- y de respuesta múltiple.

Las **preguntas de respuesta única** permiten escoger sólo una respuesta por medio de un botón de opción. Cada respuesta (correcta o incorrecta) debe incluir una corrección. Esta corrección se mostrará al estudiante junto a cada una de sus respuestas (si el cuestionario se configura para mostrar las respuestas correctas).

Para añadir una pregunta de este tipo en nuestra base de datos seleccionamos la opción "*Adding a Multiple Choice question*", e introducimos los diferentes campos requeridos (tal y como muestra la Figura 3.45).

Figura 3.35. Formulario para agregar una pregunta de opción múltiple

- Elegimos la categoría a la que pertenecerá la pregunta.

- Introducimos un nombre descriptivo a la pregunta.

Hay que evitar poner nombres genéricos como "Pregunta 1". Este nombre identificará a la pregunta en la lista de preguntas de la página de edición del cuestionario (no se mostrará a los alumnos). Seguidamente, introducimos el enunciado de la pregunta, su puntuación en el formulario y el factor de penalización. Este factor indica qué fracción de la puntuación obtenida debería substraerse por cada respuesta errónea. Esto sólo resulta relevante si el cuestionario se

ejecuta en modo adaptativo, de forma que se permite al estudiante varios intentos de resolución de la pregunta.

El Factor de penalización debe ser un número entre 0 y 1. Un factor de penalización de 1 significa que el estudiante ha de dar la respuesta correcta al primer intento para conseguir la calificación máxima. Con un factor de penalización de 0 es posible conseguir la máxima calificación independientemente del número de intentos. Finalmente, debemos seleccionar si son válidas una sola respuesta o varias. Todo este proceso se muestra en la Figura 3.46.

Image to display No images have been uploaded to your course yet
Default question grade* 1
Penalty factor* 0.1
General feedback
Path: body
One or multiple answers? One answer only
Shuffle the choices?
Number the choices? a., b., c., ...
Question name* Carpetas fundamentales en opensimulador

Figura 3.36. Configuración de una pregunta de opción múltiple

- Los siguientes pasos se deben realizar para cada una de las opciones que vaya a tener la pregunta.

Añadimos, en primer lugar, una de las posibles soluciones a la pregunta (en este caso, "*/bin y Region*"). Indicamos a continuación su calificación (en tanto por ciento). Dependiendo del número de respuestas correctas asignaremos una calificación u otra. Una sola respuesta correcta tendría una calificación del 100%, mientras que tres respuestas correctas tendrían una calificación del 33,333% (cada una de ellas). También se pueden asignar calificaciones negativas para las respuestas incorrectas. La respuesta "/bin y Region" es, en este caso, la correcta. Elegimos el porcentaje "100%", pues sólo va a ser ésta la respuesta correcta. Si la pregunta va a tener más opciones de las que Moodle muestra por defecto, haremos clic en el botón correspondiente tantas veces como necesitemos:

Blanks for 3 More Choices

La Figura 3.47 muestra el proceso descrito en el párrafo anterior.

Choice 1

Answer /bin y Region.

Grade 100 %

Feedback

Path:

Choice 2

Answer Region y carpen

Grade None

Feedback

Path:

For any incorrect response

Has respondido mal.

Path: body

Overall Feedback

For any correct response

Buena respuesta

Path: body

For any partially correct response

Has respondido parcialmente bien.

Path: body » p

Blanks for 3 More Choices

Figura 3.37. Introducción de las respuestas de una pregunta de opción múltiple

- Para finalizar pulsaremos en el botón *Guardar cambios*. A continuación veremos la ventana mostrada en la Figura 3.48.

Info | Results | Preview | Edit

Quiz | Questions | Categories | Import | Export

Question bank

Category: Categoria 1ª del tema 1. (1)

☑ Display questions from sub-categories too
☐ Also show old questions
☐ Show question text in the question list

Preguntas fáciles del Tema 1.

Create new question: Choose...

Sort by type, name

Action	Question name	Type
☐	Carpetas fundamentales en opensimulador	

Select all / Deselect all

With selected:

Delete | Move to >> | Categoria 1ª del tema 1. (1)

Figura 3.38. Visualización de la base de datos de preguntas

- Hacemos clic en el icono , para visualizar la pregunta introducida (Figura 49).

Preview Carpetas fundamentales en opensimulador

Quiz: Quiz Tema 1. Descargas y carpetas.

1 ¿Cuales son las carpetas fundamentales de OpenSim?

Marks: --/1

Choose one answer.

- a. \bin y carpen
- b. Region y carpen
- c. /bin y Region

Submit

Submit page | Submit all and finish

Fill with correct | Start again | Close preview

Figura 3.39. Previsualización de la pregunta introducida en la base de datos

La Figura 3.50 muestra qué sucedería al seleccionar una opción correcta o incorrecta para la pregunta introducida.

1 ¿Cuales son las carpetas fundamentales de OpenSim?

Marks: 0/1

Choose one answer.
- a. \bin y carpen
- b. Region y carpen ✗
- c. /bin y Region

Has respondido mal.

Submit

Incorrect

Marks for this submission: 0/1. This submission attracted a penalty of 0.1.

Quiz: Quiz Tema 1. Descargas y carpetas.

1 ¿Cuales son las carpetas fundamentales de OpenSim?

Marks: 0.9/1

Choose one answer.
- a. \bin y carpen
- b. Region y carpen
- c. /bin y Region ✓

Buena respuesta

Submit

Correct

Marks for this submission: 1/1. With previous penalties this gives **0.9/1**.

Submit page | Submit all and finish

Fill with correct | Previous state | Start again | Close preview

Figura 3.40. Proceso de respuesta a las preguntas introducidas en un formulario

3.7.2.3. Agregar preguntas a cuestionario

Una vez hemos creado las preguntas, necesitamos agregarlas al cuestionario.

- Seleccionamos la opción "*Edit / Quiz*":



- Con ello, visualizaremos la pantalla mostrada en la Figura 3.51.

Question bank

Category Default for Quiz Tema 1. Descargas y carpetas.

Display questions from sub-categories too

Also show old questions

Show question text in the question list

The default category for questions shared in context 'Quiz Tema 1. Descargas y carpetas.'.

Create new question Choose...

Sort by type, name

Action Question name Type

Carpetas fundamentales en opensimulador

Open.ini.Example

Select all / Deselect all

With selected:

Add to quiz Delete Move to >>

Default for Quiz Tema 1. Descargas y carpetas.

Add 1 random questions Add

Figura 3.41. Visualización de la base de datos de preguntas

Pulsando sobre el icono « se agregan preguntas individuales al cuestionario.

Carpetas fundamentales en opensimulador

También podemos seleccionar varias preguntas marcando las casillas que aparecen a la izquierda de la lista de las preguntas y pulsar seguidamente el botón “*Añadir a cuestionario”* que figura al final de la lista: así se agregarán todas a la vez.

Add to quiz

Si deseamos agregar todas las preguntas, utilizaremos el enlace *Seleccionar todo.*

Select all

- Seguidamente veremos el cuadro mostrado en la Figura 3.52.

Questions in this quiz

Order	#	Question name	Type	Grade	Action
↓	1	Carpetas fundamentales en opensimulador		1	»
↑	2	Open.ini.Example		1	»

Total: **2**

Maximum grade: 10

Save changes

Show page breaks

Show the reordering tool

Go

Figura 3.42. Edición de las preguntas de un formulario

Una vez que hayamos agregado una pregunta al cuestionario, ésta aparecerá en el lado izquierdo de la pantalla en la lista preguntas del cuestionario. La pregunta sigue siendo seleccionable a la derecha, pero sólo podemos agregarla al cuestionario una vez. Si seleccionamos la misma pregunta de la lista de categorías otra vez para agregarla al cuestionario, no sucederá nada.

Una vez que hayamos agregado las preguntas al cuestionario, podemos cambiar el orden de aparición en el formulario haciendo clic en los iconos en forma flecha, ↑ ↓, en el lado izquierdo de la lista de las preguntas incorporadas. Igualmente, podemos quitar una pregunta del cuestionario utilizando el icono ».

Order	#	Question name	Type	Grade	Action
↓	1	Carpetas fundamentales en opensimulador		1	»

También podemos asignar la puntuación en cada pregunta en la columna *Calificación* del cuestionario, fijando su peso en el cuestionario.

Grade
1
1

- Debemos fijar también la calificación máxima del cuestionario completo:

Maximum grade: 10

Este valor no tiene porqué ser igual a la suma de las calificaciones de las preguntas individuales. Las calificaciones alcanzadas por los estudiantes serán extrapoladas según la calificación máxima.

- <u>Después de ajustar las calificaciones de las preguntas individuales debemos hacer clic en *Guardar cambios*</u>:

Save changes

Moodle proporciona una herramienta para facilitar la ordenación de las preguntas del cuestionario que se activa mediante la casilla de verificación "*Mostrar herramienta de reordenación*".

☐ Show the reordering tool

Esta herramienta muestra campos de número de línea en la cabecera de la lista de preguntas. Estos números de línea van de 10 en 10, de modo que queda espacio para poder insertar más preguntas.

Order		#	Question name
10	↓	1	Carpetas fundamentales en opensimulador
20	↑	2	Open.ini.Example

Podemos seguidamente reordenar las preguntas cambiando los números de línea y haciendo clic en *Guardar cambios*.

- <u>Vamos a introducir en este caso el valor "5" en el campo "*Order*"</u>:

Order		#	Question name
10	↓	1	Carpetas fundamentales en opensimulador
5	↑	2	Open.ini.Example

- <u>Pulsando seguidamente el botón "*Save changes*", visualizaremos el siguiente cuadro:</u>

Order		#	Question name
10	↓	1	Open.ini.Example
20	↑	2	Carpetas fundamentales en opensimulador

Las preguntas se ordenarán de acuerdo a los números de línea especificados. También se asignan números de línea a los *saltos de página*, lo que permite moverse entre páginas del mismo modo que se ha especificado.

☐ Show page breaks

- <u>Seleccionamos la opción "Show page breaks"</u>:

☑ Show page breaks
Repaginate with 10 questions per page

- <u>Con ello, visualizaremos el siguiente cuadro</u>:

Order	#	Question name
10	1	Open.ini.Example
20	2	Carpetas fundamentales en opensimulador
30		Page break

Si no hemos marcado la opción "*Mostrar saltos de página*", no veremos los saltos de página y los números correspondientes no se mostrarán.

- Pulsamos el botón "*OS_1*".
- Seleccionamos el role "*Student*" en el combo de "*Switch role to*":

Switch role to...

Student

- Hacemos clic en el enlace "*Quiz Tema 1. Descargas y carpetas*":

Quiz Tema 1. Descargas y carpetas.

- Con ello, visualizaremos la siguiente ventana:

Quiz Tema 1. Descargas y carpetas.

Este cuestionario valora vuestros conocimientos sobre el Tema 1. Descargas y carpetas. Habrá 3 respuestas, de las cuales, **solamente 1 será la correcta**.

Grading method: Highest grade

Attempt quiz now

- Pulsando el botón "Attempt quiz now", se nos mostrará la siguiente pantalla:

Quiz Tema 1. Descargas y carpetas. - Attempt 1

1
Marks: --/1
Choose one answer.
a. Open.ini
b. ini.Example
c. Open.Example
Submit

2
Marks: --/1
¿Cuales son las carpetas fundamentales de OpenSim?
Choose one answer.
a. \bin y carpen
b. /bin y Region
c. Region y carpen
Submit

Save without submitting | Submit page | Submit all and finish

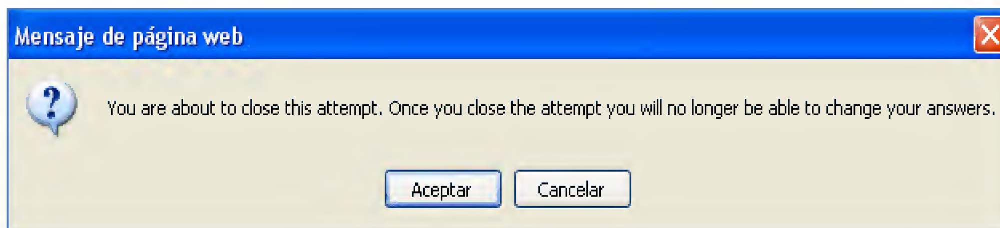
- Seleccionamos la opción "*a*" de la primera pregunta, y pulsamos el botón "*Submit*".

1
Marks: --/1
Choose one answer.
a. Open.ini
b. ini.Example
c. Open.Example
Submit

- Con ello, se nos mostrará el siguiente cuadro:

1
Marks: 1/1
Choose one answer.
a. Open.ini ✓
b. ini.Example
c. Open.Example
Submit
Correct
Marks for this submission: 1/1.

- Pulsamos al botón "*Submit all and finish*", aceptando la advertencia que se nos muestra a continuación.



- Veremos entonces la pantalla mostrada en la Figura 3.53, en la que se nos muestra el resultado de la corrección del formulario.

Quiz Tema 1. Descargas y carpetas.

Review of attempt 2

Finish review

Started on	Tuesday, 11 August 2009, 08:31 PM
Completed on	Tuesday, 11 August 2009, 08:31 PM
Time taken	-
Marks	1/2
Grade	**5** out of a maximum of 10 (**50**%)

1

Marks: 1/1

Choose one answer.

a. Open.Example ✗

b. Open.ini ✓

c. ini.Example ✗

Correct

Marks for this submission: 1/1.

2

¿Cuales son las carpetas fundamentales de OpenSim?

Marks: 0/1

Choose one answer.

a. \bin y carpen ✗

b. Region y carpen ✗

c. /bin y Region ✓

Has respondido mal.

Incorrect

Marks for this submission: 0/1.

Finish review

Figura 3.43. Visualización de la corrección de un formulario

- Tras pulsar el botón "*Finish review*", podremos visualizar el resultado de todos los intentos previos (Figura 3.54).

Summary of your previous attempts

Attempt	Completed	Marks / 2	Grade / 10
1	Tuesday, 11 August 2009, 08:30 PM	1	5
2	Tuesday, 11 August 2009, 08:31 PM	1	5

Highest grade: 5 / 10.

Re-attempt quiz

Figura 3.44. Resumen del número de intentos de un formulario

En este caso, las estadísticas muestran que hemos completado dos veces el test, y en ambas hemos obtenido una nota de 5 sobre 10.

- Si pulsamos en "*Re-attempt quiz*" podemos volver a intentar el test, hasta completarlo satisfactoriamente:

Summary of your previous attempts

Attempt	Completed	Marks / 2	Grade / 10
1	Tuesday, 11 August 2009, 08:30 PM	1	5
2	Tuesday, 11 August 2009, 08:31 PM	1	5
3	Tuesday, 11 August 2009, 08:37 PM	2	10

Highest grade: 10 / 10.

Re-attempt quiz

3.8. Insertar una imagen

El editor de texto HTML proporciona una interfaz muy parecida a la de un procesador de textos para generar de una forma cómoda el código HTML correspondiente a los elementos insertados junto al texto (imágenes, tablas, caracteres especiales…).

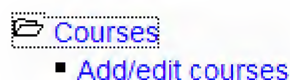
Este editor de texto debe habilitarlo el administrador del sitio Moodle. Cuando el editor de texto está activado se muestra una barra de herramientas (Figura 3.55).



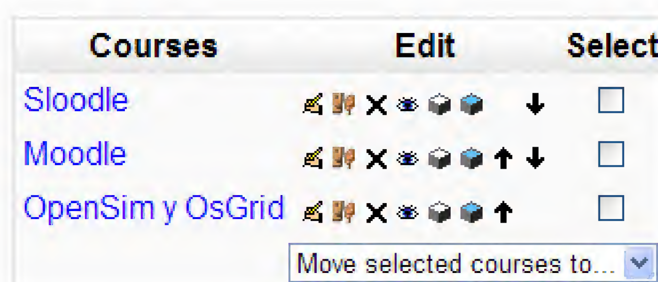
Figura 3.45. Barra de herramientas para la edición de imágenes

Es importante tener en cuenta que cuando insertamos una imagen en un documento Web, ésta realmente no se inserta, sino que lo que realmente se inserta es un enlace al archivo de imagen.

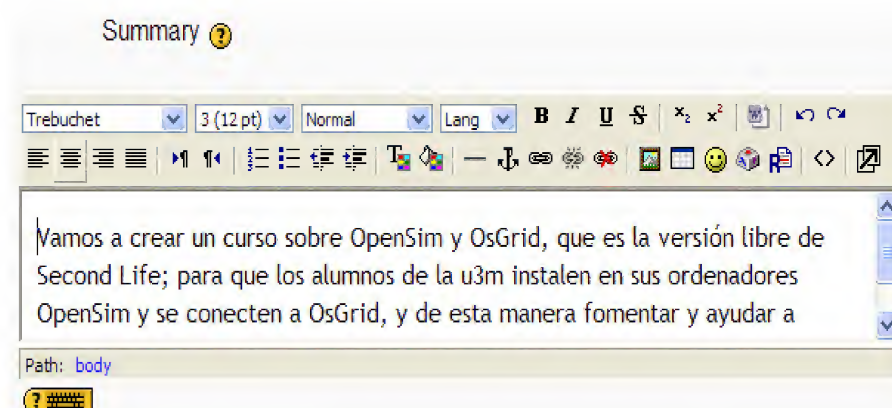
- Accedemos a la opción:



- Visualizamos con ello una ventana con los cursos que tenemos disponibles actualmente.



- Hacemos clic sobre el icono de "OpenSim y OsGrid" y visualizamos la siguiente ventana, en la opción de "Summary":



- Este archivo (imagen) debe subirse previamente al servidor Moodle.

De este modo, cuando pulsamos sobre el botón se abre una ventana, Figura 3.56, que muestra el contenido de la carpeta del curso donde se encuentran todos los archivos subidos hasta el momento y permite subir nuevos archivos.

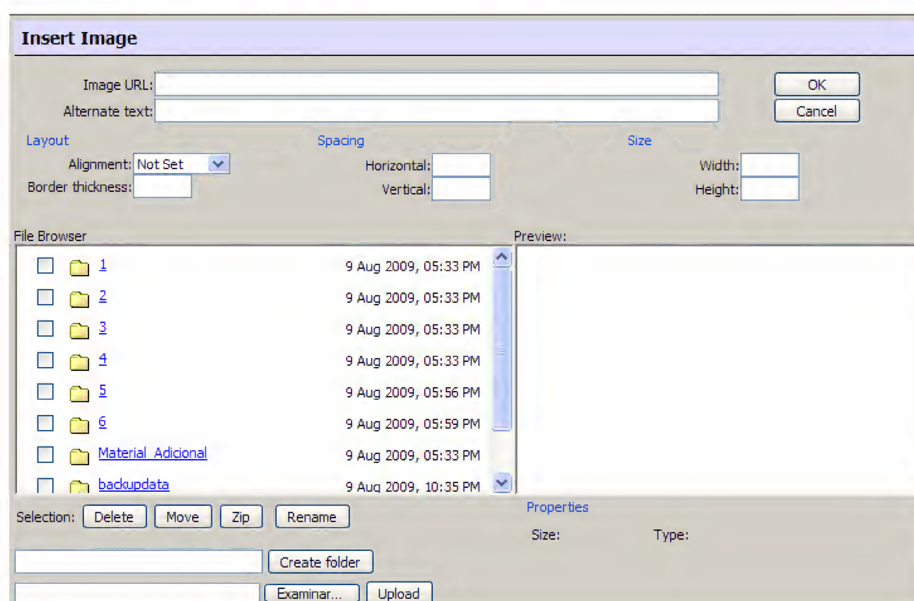
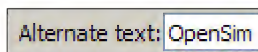


Figura 46. Visualización del total de archivos y directorios en nuestro curso

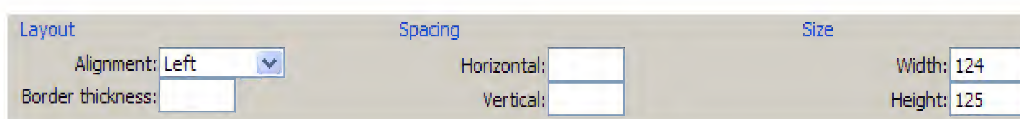
Si la imagen que queremos insertar se encuentra en el servidor tan solo tendremos que seleccionarla en la ventana de la izquierda *"Navegador de archivos"*. Una vista previa aparecerá en la ventana de la derecha "*Previsualizar*", y se mostrará su tamaño en los cuadros *Anchura* y *Altura.*

Si la imagen no está cargada en el servidor, pulsaremos el botón "*Examinar... "* para localizar el archivo en nuestro equipo. Seleccionando la opción "*Upload*" podremos subirla al servidor.

- Introducimos un texto alternativo (Alternate text), por si no se ve la imagen:

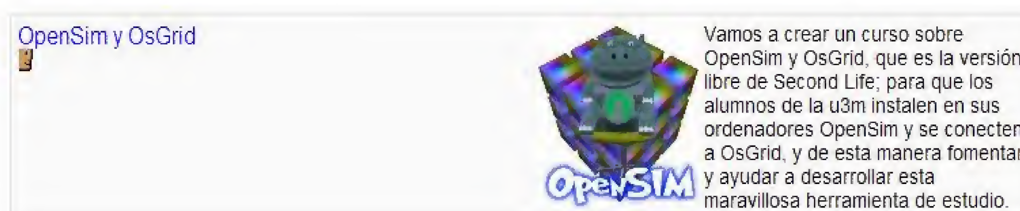


- Opcionalmente, podemos definir el formato en el que se insertará la imagen con el texto, seleccionando para ello una de las posibles opciones en la lista desplegable *Alinear*. Podemos además insertar un recuadro alrededor de la imagen indicando el grosor del borde en el cuadro de texto *Borde* y ajustar la separación entre el texto y la imagen, escribiendo la separación en los cuadros de texto *Horizontal* y *Vertical*.



- Una vez realizados todos los ajustes, pulsaremos en el botón *OK* para insertar la imagen en el documento.

- Seguidamente, hacemos clic en el botón "Save changes", acudimos al apartado *Cursos* y visualizamos el resultado.



3.9. Consultas

La consulta es una actividad sencilla que permite al profesor plantear una pregunta, especificando las posibles respuestas entre las cuales deben elegir los estudiantes. Los resultados de la consulta se pueden ver de forma intuitiva en forma de diagrama de barras con las imágenes de los estudiantes. El profesorado puede evitar que se vea qué opción eligió cada estudiante en aquellas preguntas que traten aspectos más personales.

3.9.1. Características de una consulta

- Se pueden asimilar como votaciones. Pueden usarse, de este modo, para votar sobre una determinada cuestión o para recibir el consentimiento de los estudiantes sobre un determinado punto.

- El profesorado puede ver una tabla que presenta de forma intuitiva la opción elegida por cada estudiante.

- Se puede permitir que los alumnos/as vean la tabla anterior o un gráfico anónimo y actualizado de los resultados.

- Esta consulta no es científica, responde tan sólo a las respuestas voluntarias de los participantes que deseen exponer su opinión.

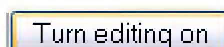
- Podemos limitar el número de máximo de respuestas por opción.

- Podemos descargar el resultado de la consulta en formato *.txt* o *.xls* y generar otros gráficos más descriptivos desde una hoja de cálculo.

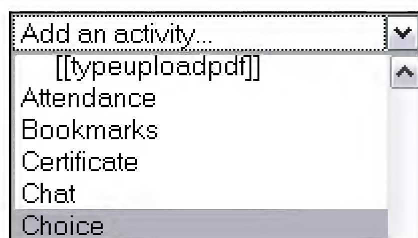
3.9.2. Cómo crear una consulta

Para crear una consulta debemos seguir el siguiente proceso.

- Activamos el modo de edición:



- Accedemos a la lista desplegable *Agregar una actividad*, para añadir la consulta en la sección que nos interese.



Configuraremos en esta pantalla los diferentes parámetros que controlan su funcionamiento.

- Título de la consulta.

  Es el nombre identificativo con el que quedará enlazada la consulta en la página principal del curso, dentro de la sección elegida. Se pueden incluir marcas o etiquetas HTML para resaltar parte o la totalidad del nombre. En este caso, escribimos "*Fecha de examen del Tema 1*" en el campo "*Choice name*".

  Choice name* Fecha de examen del Tema 1

- Texto de la consulta.

  Es el enunciado o descripción de la pregunta planteada. Debe estar formulada de manera clara y concisa.

Choice text*

Trebuchet | 1 (8 pt) | Lang

Consultamos a los alumnos cuando proponen que sea la fecha del primer examen de OpenSim de tema 1.

- Limitar el número de respuestas permitidas.

  Por defecto, cualquier estudiante puede elegir cualquiera de las opciones que se le presentan. Pero si habilitamos esta opción (*Permitir*), podremos limitar el número de respuestas para cada opción de manera independiente.

  Este límite se establece con el parámetro *Límite* asociado a cada respuesta. Una vez alcanzado este límite nadie más podrá seleccionar esa opción. Por el contrario, si inhabilitamos este parámetro, cualquier número de participantes podría seleccionar cualquiera de las opciones.

Limit

Limit the number of responses allowed: Disable

Limit: 0

- Consulta *1* – Consulta *n*.

  Son las opciones de respuesta posibles a la pregunta planteada. Rellenamos los valores de cuatro consultas:

Choice 1

Choice: 20 de enero

Choice 2

Choice: 23 de enero

Choice 3

Choice: 24 de enero

Choice 4

Choice: 28 de enero

Si requerimos más campos, haremos clic sobre el botón [Add 3 fields to form].

- Limitar la validez temporal de las respuestas.

  Esta opción permite definir el período de tiempo en el que los participantes pueden responder a la consulta.

Restrict answering to this time period

Restrict answering to this time period

Open 14 August 2009 06 40

Until 14 August 2009 06 40

No restringimos en nuestro caso la respuesta a un periodo de tiempo.

- Modo mostrar.

  Nos permite elegir la forma (vertical u horizontal) en la que se mostrarán los resultados a los participantes. La elección de una u otra forma podría vincularse al número y longitud de las respuestas ofrecidas a los estudiantes (por ejemplo, si son muchas, la forma más adecuada de presentación sería la vertical).

  Elegimos la opción "Display horizontally":

Display Mode Display horizontally

- Publicar resultados.

  Nos permite determinar si los participantes pueden ver los resultados de la consulta y cuándo son visibles:

  - *No mostrar los resultados a los alumnos*: no se presentan los resultados a los estudiantes en ningún momento. Sólo los puede ver el docente.

  - *Mostrar los resultados al estudiante después de su respuesta*: sólo se muestran los resultados acumulados si el estudiante que accede ya ha votado en esta consulta.

  - *Mostrar los resultados a los estudiantes sólo después de cerrar la consulta*: los resultados sólo se muestran una vez cerrada la consulta, bien porque ha expirado el plazo o bien porque han votado ya todos los participantes.

  - *Mostrar siempre los resultados a los estudiantes*: se muestran los resultados acumulados a cualquier estudiante y en cualquier momento, haya o no votado.

  Elegimos la opción "*Show results to students after they answer*":

Publish results Do not publish results to students

Do not publish results to students
Show results to students after they answer
Show results to students only after the choice is closed
Always show results to students

- Privacidad de los resultados.

  Establece si los resultados se publicarán incluyendo o no información sobre la opción elegida por cada usuario, es decir, si la consulta será anónima o no. Es muy

recomendable usar la opción de consulta anónima cuando se traten contenidos que requieran privacidad.

Elegimos consulta anónima:

Privacy of results | Publish anonymous results, do not show student names

- Permitir la actualización de la consulta.

  Si seleccionamos *Sí*, los participantes podrán cambiar su opción en cualquier momento.

  Elegimos que no se puede actualizar la consulta:

  Allow choice to be updated | No

- Mostrar columna con las preguntas no respondidas.

  Si seleccionamos *Sí*, se muestra una columna con los participantes que no han respondido a la consulta.

  Elegimos que no muestre la columna con las no respondidas:

  Show column for unanswered | No

- Modo de grupo.

  Establece la forma de interacción entre los miembros de los grupos de trabajo.

  Elegimos que no queremos grupos:

  Group mode | No groups

- Visibilidad.

  Permite mostrar u ocultar la actividad a los estudiantes.

  Elegimos que mostramos la actividad de los estudiantes:

  Visible | Show

- Pulsaremos en el botón *Guardar cambios* y la consulta estará lista para su uso.

  Save and return to course

  Fecha de examen del Tema 1

3.9.3. Participar en la consulta

Como en las demás actividades, el acceso a una consulta se puede hacer desde la página principal del curso haciendo clic sobre la consulta que deseamos ver.

Fecha de examen del Tema 1

- En la parte superior de la página se visualiza el enunciado de la pregunta.

Consultamos a los alumnos cuando proponen que sea la fecha del primer examen de OpenSim de tema 1.

Justo debajo veremos las posibles respuestas (un botón de opción para cada una de ellas).

20 de enero 23 de enero 24 de enero 28 de enero

- Para participar, el estudiante tan solo tendrá que seleccionar su respuesta y pulsar el botón *Guardar mi elección*.



23 de enero

Guardar mi elección

- Tras participar en la consulta, se muestra el estado actual de la encuesta (Figura 3.57).

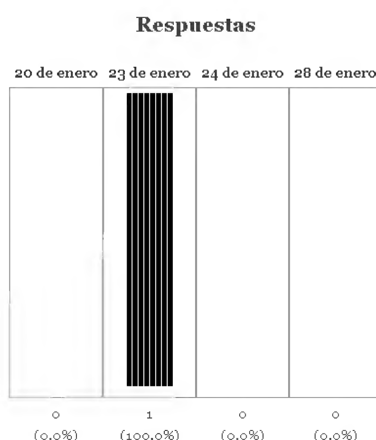


Figura 3.47. Visualización de las estadísticas de una consulta

3.9.4. Gestión de una consulta

El profesorado siempre tiene un control sobre la evolución de las consultas. Accediendo a la consulta, puede ver los resultados de la misma y los estudiantes que eligieron cada una de las respuestas.

En la parte superior derecha de esa página se encuentra un enlace que permite *Ver n respuestas*, donde *n* es el número de respuestas realizadas.

View 1 responses

Download in ODS format | Download in Excel format | Download in text format

Se mostrará la página de los informes con las opciones de la consulta, compuesto con las fotografías de las personas que han contestado y, opcionalmente, una barra con las que aún no han contestado.

Para obtener un informe debemos pulsar el botón según el formato de nuestra elección (ODS, Excel o de texto).

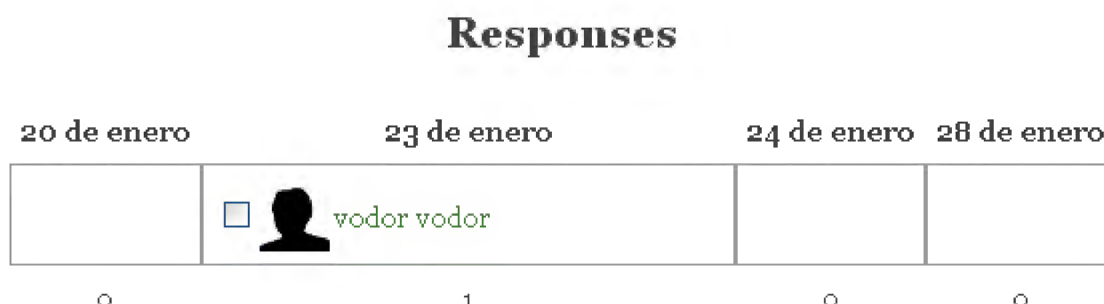


Figura 48. Visualización de un informe de una consulta

También tenemos la posibilidad de seleccionar y eliminar las respuestas deseadas de una consulta.

Select all / Deselect all With selected

Para ello, debemos seleccionarlas previamente



y luego elegir la opción *Borrar* en la lista desplegable de la parte inferior.

Deselect all

3.10. Tareas

Una tarea es una actividad de Moodle que permite al profesorado asignar un trabajo a los estudiantes que éstos deberán preparar normalmente en un formato electrónico (documento de texto, presentación electrónica, imagen gráfica, vídeo, archivo fuente en un determinado lenguaje, etc.) y remitirlo, subiéndolo al servidor. Los documentos quedarán almacenados para su posterior evaluación a la que podrá añadirse un comentario que llegará de forma independiente al estudiante mediante correo electrónico.

Existen cuatro tipos de tareas:

- Subida avanzada de archivos.

  Este tipo de tarea se utiliza cuando se solicita al estudiante que entregue más de un archivo. Normalmente, el número de archivos que debe subir se especificará en la descripción de la tarea.

➢ Texto en línea.

Permite al estudiante hacer uso del editor de texto HTML para escribir directamente el trabajo solicitado.

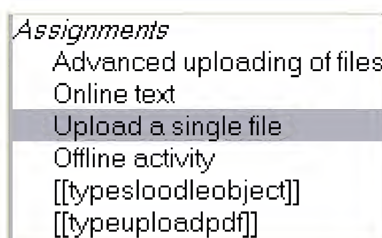
➢ Subir un único archivo.

En este caso, se permite que los estudiantes puedan subir un archivo de cualquier tipo. Éste podría ser un documento realizado con un procesador de textos, o una imagen, un sitio web comprimido o cualquier otro formato que se les haya requerido. El profesorado puede calificar online las tareas remitidas de este modo.

➢ Actividad offline.

Este tipo de tarea es útil cuando el trabajo se realiza fuera de la plataforma y, además, lo deben realizar en un formato no electrónico como, por ejemplo, una lámina de dibujo. Los estudiantes pueden ver una descripción de la tarea, pero no pueden subir archivos. No obstante, los profesores/as pueden calificar a todos los estudiantes y los estudiantes recibirán notificaciones de sus calificaciones.

- Seleccionamos *Upload a single file*:



3.10.1. Características de las tareas

Las características de las tareas son las siguientes:

➢ Puede especificarse la fecha final de entrega de una tarea, que aparecerá en el calendario del curso, y la calificación máxima o el uso de escalas personalizadas que se le podrá asignar.

➢ Los estudiantes pueden subir sus tareas (en cualquier formato de archivo) al servidor. Se registra la fecha en que se han subido.

➢ Se permite enviar tareas fuera de tiempo, pero el profesor/a puede ver claramente el tiempo de retraso.

➢ En cada tarea en particular, puede evaluarse a toda la clase (calificaciones y comentarios) en una única página con un único formulario.

➢ Las observaciones del profesor/a se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.

➢ El profesorado tiene la posibilidad de permitir el reenvío de una tarea tras su calificación (para volver a calificarla).

- Se permite el uso de tareas no en línea, donde en lugar de un archivo digital, la tarea es una actividad externa como por ejemplo recitar un poema o realizar una actividad físico-deportiva.

- Se permite el uso de tareas en línea de texto. Son como los diarios de las versiones anteriores a la 1.5 de Moodle. Cumplen la función de un cuaderno de clase.

3.10.2. Creación de una tarea

La primera parte del proceso de creación de una tarea es común a todos los tipos de tareas.

- Recordemos que para crear una nueva tarea hay que activar el modo de edición:

Turn editing off

- A continuación, seleccionamos el tipo de tarea de la lista desplegable *Agregar una actividad* de la sección que nos interese.

Upload a single file

Esta acción nos conduce al formulario de configuración de la misma, donde completaremos los campos que se solicitan. El primer bloque de parámetros, correspondientes a la sección de ajustes generales, es común a todos los tipos de tareas.

- Título de la tarea.

  Es el texto identificativo con el que quedará enlazada la tarea en la página principal del curso, dentro de la sección elegida. Se pueden incluir marcas o etiquetas HTML para resaltar parte o la totalidad del nombre.

  Asignamos un nombre a una tarea: "Descargar OpenSim".

Assignment name* Descargar Opensim.

- Descripción.

  En este cuadro se plantea el enunciado y descripción del trabajo que deben realizar los estudiantes. También podemos especificar sus objetivos, los apartados que deben tratarse, algunas instrucciones para realizar la tarea correctamente (por ejemplo, tamaño y formato de los archivos a subir), etc.

  El sistema añade automáticamente a esta descripción, la calificación máxima asignada a la tarea y la fecha límite de entrega de la misma (indica también cuántos días/horas faltan para el vencimiento del plazo).

  Describimos la tarea:

Description*

Trebuchet 1 (8 pt) Lang

Los alumnos deberan Descargarse el programa OpenSim. Y tendran que subir un archivo con sus propias anotaciones.

Path: body

- Calificación.

  Nos permite escoger entre varias escalas de calificación, tanto cuantitativas como cualitativas.

  Elegimos una calificación sobre 10:

  Grade 10

- Disponible en.

  Especifica la fecha a partir de la cual la tarea está disponible para los estudiantes. Para fijar una fecha debe estar desactivada la casilla *Inhabilitado*.

  Available from 17 August 2009 09 05 Disable

- Fecha límite de entrega.

  Determina el día y la hora en que acaba el plazo para entregar los trabajos solicitados. Moodle establece un período "de gracia" de 30 minutos tras la hora de cierre.

  Due date 24 August 2009 09 05 Disable

- Impedir envíos retrasados.

  El profesorado puede decidir si permite al estudiante enviar sus trabajos una vez que ha expirado la fecha de entrega de la tarea.

  Establecemos que permitimos envíos retrasados:

  Prevent late submissions Yes

El siguiente grupo de parámetros son específicos de algunas tareas (pueden aparecer en el formulario de configuración de un tipo de tareas y no en otros).

- Permitir reenvío.

  Por defecto, los estudiantes no pueden reenviar las tareas después de que han sido calificadas. Si activamos esta opción, se permitirá a los estudiantes reenviar las tareas después de que hayan sido calificadas (con el objeto de volver a calificarlas). Esto puede ser útil si el docente quiere animar a los estudiantes a hacer un mejor trabajo en un proceso iterativo. Obviamente, esta opción no es aplicable para las tareas *Offline*.

  Permitimos el reenvío:

  Allow resubmitting ? Yes

- Alertas de email a los profesores.

  Si se activa esta opción, el profesorado recibirá una alerta mediante un breve correo siempre que los estudiantes añadan o actualicen el envío de una tarea.

  Sólo se avisará al profesorado con permiso para calificar ese envío en particular. De este modo, si, por ejemplo, el curso usa grupos separados, entonces los docentes asignados a grupos específicos no recibirán información sobre los estudiantes que pertenecen a otros grupos. Por supuesto, nunca se envía correo cuando las actividades son fuera de línea, toda vez que en ese caso los estudiantes no realizan envíos.

  Configuramos que queremos que le se envíe un e-mail al profesor:

  Email alerts to teachers ? Yes

- Tamaño máximo.

  Sirve para determinar el tamaño máximo de los archivos que subirán los estudiantes.

  Establecemos el tamaño máximo en 2MB:

  Maximum size 2MB

Este último bloque de ajustes comunes del módulo aparece en todos los tipos de tareas:

- Modo de grupo.

  Establece la forma de interacción entre los miembros de los grupos de trabajo (cada estudiante sólo puede pertenecer a un único grupo). Tenemos las siguientes opciones:

  - *No hay grupos*: todos los estudiantes formarán parte de un único grupo.
  - *Grupos separados*: en el que los grupos trabajarán independientemente, es decir, los estudiantes de un grupo sólo pueden ver a su propio grupo, siendo los demás grupos invisibles para ellos.

- *Grupos visibles*: cada estudiante trabajará dentro de su grupo, pero podrá ver el trabajo de otros grupos.

El modo grupo puede ser definido a dos niveles:

1. *Nivel Curso*:

   El modo grupal definido a nivel de curso viene por defecto para todas las actividades definidas dentro del curso.

2. *Nivel Actividad*:

   Toda actividad que soporte grupos puede definir su propio modo de agrupación. Si el curso está configurado como forzar modo de grupo entonces no se tendrá en cuenta la configuración de cada actividad.

Elegimos que no queremos grupos:

Group mode No groups

- Visible.

Permite mostrar u ocultar el cuestionario a los estudiantes. Elegimos visible:

Visible Show

3.10.3. Gestión de una tarea

3.10.3.1. Subir una tarea

Una vez configurada la tarea, los estudiantes podrán leer el enunciado y comenzar a enviar sus trabajos. Para ello, deberán seleccionar un archivo de su PC.

- Hacemos clic en el siguiente enlace:

Descargar carpetas

- A través de un formulario incluido en la propia tarea podemos realizar el envío de ficheros. Pulsamos, para ello, el botón *Examinar y seleccionamos un fichero que hayamos creado con la explicación de la tarea.*

Upload a file (Max size: 2MB)

Examinar...

- Tras pulsar el botón *Subir este archivo*, se nos indicará que la tarea se ha subido correctamente al servidor:

Upload this file

File uploaded successfully

Continue

3.10.3.2. Revisar las tareas

Podemos revisar los trabajos de los estudiantes pulsando sobre el enlace a la tarea desde la página principal del curso, en la columna *Enviada* o mediante un enlace que aparece en la parte superior derecha de la tarea.

- Nos muestra en nuestro sitio Web que ninguna tarea ha sido subida:

No attempts have been made on this assignment

- Hacemos clic en este enlace, y se nos muestra la siguiente información:

See all course grades

First name : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Surname : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

First name / Surname	Grade	Comment	Last modified (Student)	Last modified (Teacher)	Status	Final grade
vodor vodor	-				Grade	-

Submissions shown per page 10

Allow quick grading

Save preferences

Figura 3.49. Visión de las tareas por parte de los estudiantes

Ahora, haremos que nuestro alumno "Vodor" entregue su tarea:

- Vodor introduce los datos de su cuenta para acceder al curso OpenSim y OsGrid. Seguidamente, hace click en el enlace *Descargar Carpetas* y sube el fichero correspondiente a su tarea.

Nombre de usuario vodor

Contraseña •••••• Entrar

OpenSim y OsGrid

Descargar carpetas

Subir un archivo (Tamaño máximo: 2Mb)

Examinar...

- Tras subir dicho fichero, el sistema permite la posibilidad de reenvíos.

Archivo subido con éxito

Continuar

Trabajo_de_Vodor_de_Tema_1_sobre_la_tarea_Descargas_y_Carpetas.doc

Subir un archivo (Tamaño máximo: 2Mb)

Examinar...

Subir este archivo

- Ahora iniciamos sesión como administrador:

Nombre de usuario 100048464
Contraseña •••••••••• Entrar

- Hacemos clic en el curso *OpenSim y OsGrid*, seguidamente en la tarea *Descargar Carpet*as y, por último, en el enlace *View 1 submitted assignments*.

OpenSim y OsGrid

Descargar carpetas

View 1 submitted assignments

- Nos muestra la siguiente ventana:

See all course grades

First name : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Surname : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

First name / Surname	Grade	Comment	Last modified (Student)	Last modified (Teacher)	Status	Final grade
vodor vodor	-		Trabajo_de_Vodor_de_Tema_1_sobre_la_tarea_Descargas_y_Carpetas.doc Monday, 17 August 2009, 10:47 AM		Grade	-

Submissions shown per page 10
Allow quick grading
Save preferences

Figura 3.50. Visión de las tareas por parte de los docentes

La página de revisión a la que accedemos nos mostraría un listado de los estudiantes, el nombre del archivo que subió y en qué fecha lo hizo y, a la derecha, el enlace *Calificación (Grade)*.

- Pulsando sobre dicho enlace, se accede a la ventana de retroalimentación para ese estudiante:

Feedback:vodor vodor:Descargar carpetas - Mozilla Firefox
http://tesis.edumoot.com/mod/assignment/submissions.php?id=164&userid=9&mode=single&offset=0

Grade No grade
Final grade: -

Trebuchet 1 (8 pt) Lang

Path:

Send notification emails

Save changes Cancel

vodor vodor
Monday, 17 August 2009, 10:47 AM (6 days 22 hours early)
Trabajo_de_Vodor_de_Tema_1_sobre_la_tarea_Descargas_y_Carpetas.doc

Figura 3.51. Ventana para las respuestas personalizadas a los alumnos

- En dicha ventana encontramos un enlace al archivo subido por el estudiante, al hacer clic sobre él se nos da la posibilidad de descargarlo o de abrirlo.

Trabajo_de_Vodor_de_Tema_1_sobre_la_tarea_Descargas_y_Carpetas.doc

- Una vez corregido el trabajo, procederemos a su calificación mediante la lista desplegable *Calificación* y, opcionalmente, podemos utilizar el editor HTML para introducir nuestra valoración del trabajo y algunos comentarios sobre el mismo.

Grade 10 / 10

- También se nos muestra un detalle interesante, como es la fecha en que se subió el archivo.

(6 days 22 hours early)

Si hubiéramos configurado la tarea para permitir envíos retrasados y si, efectivamente, ese envío se realizó pasada la fecha de entrega, se mostrará el retraso correspondiente en color rojo.

La página de revisión nos muestra una tabla con el listado de los estudiantes, la fecha en que subieron los archivos, la fecha de corrección por parte del profesor/a, la calificación otorgada y los comentarios del profesorado respecto a la tarea corregida.

	First name / Surname	Grade	Comment	Last modified (Student)	Last modified (Teacher)	Status	Final grade
	vodor vodor	10 / 10		Trabajo_de_Vodor_de_Tema_1_sobre_la_tarea_Descargas_y_Carpetas.doc Monday, 17 August 2009, 10:47 AM	Monday, 17 August 2009, 11:02 AM	Update	10.00

Figura 3.52. Ventana de revisión de las tareas

Podemos ordenar ese listado según diversos criterios simplemente haciendo clic sobre el nombre de cada columna. Por ejemplo, si queremos ordenar los estudiantes de menor a mayor calificación, haremos clic sobre el título *Calificación*. Si hacemos un nuevo clic sobre ese mismo título, cambiaremos la ordenación de mayor a menor nota. También podemos ocultar columnas mediante una casilla que aparece junto al título de la columna.

Conclusiones Moodle

Moodle es un sistema experto construido mediante la disciplina de la Ingeniería del Conocimiento, basándose todo su desarrollado en la experiencia de profesores, de tal forma que se trata de una herramienta excepcional para llevar a cabo aplicaciones educativas.

En este capítulo no sólo se explican las características y funcionalidades más importantes de Moodle, sino que además he desarrollado un curso sobre esta herramienta justamente utilizándola. En el curso se recoge toda la información necesaria para poder realizar las configuraciones de las herramientas que presentaremos en el siguiente capítulo describiendo Sloodle (integración de Second Life y Moodle). De esta manera, el lector de estos cursos dispone todo el material necesario para crear su propia aula virtual de e-learning en 3D.

Tras la realización de este curso de Moodle yo mismo utilizo esta herramienta para impartir mis clases, ya que me proporciona una gran ayuda a la hora de dar un mejor servicio a mis alumnos y organizar de mejor forma los contenidos de la materia a impartir.

El principal inconveniente que posee Moodle surge a la hora de albergar esta plataforma en un servidor web gratuito, pues es usual que se nos planteen muchos problemas de instalación, los cuales no son, en muchos casos, nada fáciles de resolver.

CAPÍTULO IV

SECOND LIFE

Capítulo 4

Second Life

4.1 Introducción

Second Life (abreviado como SL) es un metaverso lanzado el 23 de Junio de 2003, desarrollado por Linden Research Inc., que ha tenido una atención internacional de manera creciente desde el año 2006.

Para hacer uso de este programa es necesario crearse una cuenta en www.secondlife.com y bajar el programa visor llamado Second Life Viewer. Al registrarse y acceder sus usuarios pasan a denominarse "residentes" o de manera abreviada AV, que significa *avatars*. La manera en que los residentes interactúan a través de SL, lo que le confiere uno de los principales atractivos de este mundo virtual, es a través de estos avatares, que son personajes en 3D completamente configurables, lo que le permite a los usuarios la capacidad de "convertirse en otra persona" y gozar (como el mismo nombre del programa indica) de una segunda vida. Esta característica promueve en el mismo mundo una avanzada interacción virtual para que los residentes de SL puedan explorar el mundo, conocer a otras personas, socializarse, participar en actividades grupales de acuerdo a sus gustos, entre otras muchas cosas. Su segundo atractivo más importante es la posibilidad de crear objetos e intercambiar diversidad de productos virtuales a través de un mercado abierto que tiene como moneda local el Linden Dólar ($L).

Los principales objetivos del presente capítulo son:

- Describir las características principales que posee Second Life para crear un entorno educativo.

- Mostrar cómo crear una cuenta en Second Life para así disponer de un avatar e interactuar en este mundo virtual.

- Describir cómo crear varios enlaces desde un curso de la plataforma Moodle para poder conectarse a varias islas en Second Life, y de esta forma acceder a los diferentes contenidos disponibles en dichas islas y conseguir que el curso desarrollado sea lo más completo y organizado posible.

- Detallar el funcionamiento de la herramienta SCRATCH, utilizada para crear de forma muy sencilla scripts en Linden Scripting Language, que es el lenguaje de Second Life.

- Describir la herramienta ROBO[2], utilizada para que un bot pueda acatar las órdenes que le suministremos.

4.2. Crear una cuenta en SL

Introducimos en la barra de navegación del explorador Web la siguiente dirección: http://www.secondspain.es/ y seleccionamos el enlace "**Crear una cuenta en Second Life**" para poder disponer así de nuestro avatar. Rellenando el formulario que se nos facilita, Figura 4.1, dispondremos ya del mismo.

Nombre del avatar: eduardorojo
(El nombre debe contener entre 2-31 caracteres, exclusivamente cifras o letras)

Apellido del avatar: Aboma

Email: eduardorojosanchez2@gmail.com
(El email debe ser real, a través suyo recibirás los avisos de tus amigos en SL cuando no estés conectado)

Tu fecha de nacimiento (real): 01 Enero 1979
(No está permitido el registro en Second Life a menores de 18 años)

cuts electronicists
Type the two words: cuts electronicists
reCAPTCHA™ stop spam. read books.

Crear tu cuenta en SL

Figura 53. Formulario para crear una cuenta en Second Life

La fecha de nacimiento debe ser real, porque dependiendo de la edad que tengamos nos encontraremos en Second Life con personas mayores o menores de 18 años.

Tras remitir el formulario, SL nos facilita la información mostrada en la Figura 4.2 (entre ella, la contraseña de nuestro avatar).

Virtual Spain - Crea tu cuenta en Second Life®

Bienvenido **eduardorojo** , tu contraseña es

MUY IMPORTANTE: Guarda o imprime esta página para no perder esta contraseña, te será muy difícil recuperar tu avatar si la pierdes!

Tu avatar ha sido creado correctamente. Ahora haz click aquí para descargar el programa de Second Life (Windows, Mac OS X o Linux): una vez lo hayas descargado, **instálalo y arráncalo para entrar en Second Life usando el nombre y apellido del avatar que acabas de crear y la contraseña que te han asignado** (podrás cambiarla si quieres desde http://www.secondlife.com, mira en el apartado Gestiona tu cuenta de usuario en SL).

Ahora puedes unirte a la mayor comunidad de usuarios hispanohablantes de SL, Second Spain, para conocer a miles de personas que ya están disfrutando de SL y participar en cursos, foros, concursos, votaciones, eventos y mucho más...

Figura 54. Datos del registro completado en Second Life

4.3. Descarga e instalación del explorador de Second Life

Second Life, al igual que una web en 2D necesita un explorador web, requiere de visor 3D para navegar por el mundo virtual. Introducimos en la barra de navegación del explorador Web la siguiente URL: http://secondlife.com/support/downloads.php. De dicha dirección podemos descargar el visor de Second Life, disponible para los sistemas operativos Linux, Windows y MAC, tal y como muestra la Figura 4.3. Al iniciar la instalación de la aplicación se nos solicitará que seleccionemos el idioma.

Second Life® Installer



Windows 2000/XP/Vista:

Version Number: 1.23.4.123908
Download Size:

WIN | MAC | Linux (Beta)


Second_Life_1-23-4-123908_...

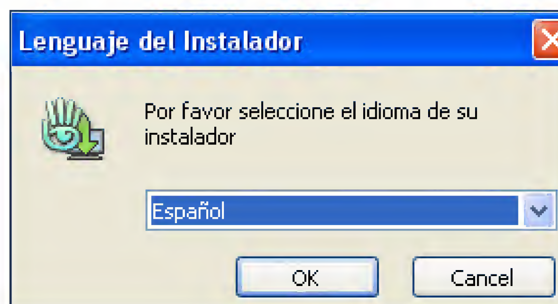


Figura 55. Descarga e instalación del visor de Second Life

Tras la instalación, dispondremos del siguiente acceso directo para acceder al programa visor de Second Life:



4.4. Acceder a Second Life

Tras hacer doble clic en el acceso directo de Second Life podremos acceder al visor de Second Life (Figura 4.5). El primer paso es introducir el nombre, apellido y contraseña de nuestro avatar, tal y como muestra la Figura 4.4.

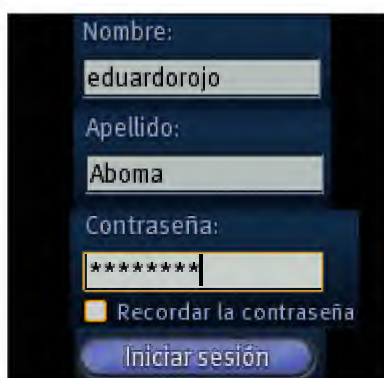


Figura 56. Inicio de sesión en Second Life



Figura 57. Explorador de Second Life

La Figura 4.6 muestra nuestro avatar en Second Life.



Figura 58. Visualización de nuestro avatar en Second Life

4.5. Análisis de Second Life

En este apartado resumimos las características principales de Second Life.

- **Comunicación por voz.**

  La comunicación por voz es opcional, pues puede ser activada o desactivada. Los propietarios de las tierras pueden controlar el uso de la voz en sus respectivas tierras.

- **Grupos. Tus comunidades.**

  Puedes acceder a grupos o creas tus propios grupos para compartir actividades e intereses, tal y como muestra la Figura 4.7. En esta figura vemos los grupos a los cuales pertenecemos.

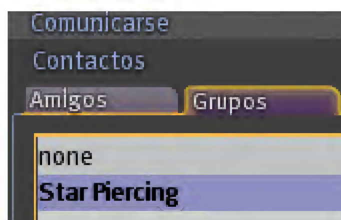


Figura 59. Grupos en Second Life

- **Teletransporte.**

  Se trata de la manera más rápida de recorrer grandes distancias.



- **Búsqueda en el Mapa.**

  Puedes abrir el mapa (Figura 4.8), buscar el nombre de una región, parcela o cualquier otra cosa que quieras, y teletransportarte o viajar hasta ese punto.



Figura 60. Mapa de Second Life

Podemos ver en el mini-mapa como está distribuida una *sim* y teletransportamos a una zona de la parcela pinchando sobre la zona y haciendo seguidamente clic en el botón teletransportarte.

- **Volar.**

Es la manera más veloz de moverse con el avatar en distancias cortas. En la Figura 4.9 podemos ver al avatar volando sobre una parcela.



Figura 61. Avatar volando en Second Life

- **Presentador de PDF, visor de vídeo, audio, Web, imágenes.**

Desde Second Life podemos:

- <u>Ver archivos PDF</u>: Para ver archivos PDF utilizamos la utilidad de Sloodle de Presenter 0’4 Rev 7.
- <u>Subir nuestros propios vídeos</u>: Para subir un vídeo al inventario de nuestro avatar hay que pagar. Esta característica es muy importante pues nos permite crear videotutoriales educativos.

Un videotutorial es un vídeo grabado mediante un programa de captura de pantalla (como, por ejemplo, Camtasia Studio), que puede incluir tanto la captura pantalla del monitor como la voz grabada mediante un micrófono. Además, pueden introducirse subtítulos de las explicaciones por si el alumno en ese momento determinado no dispusiese de altavoces del ordenador.

El principal problema de subir vídeos a nuestro *Sim*, es que los avatares que se acerquen a ella requerirán de un ordenador, una tarjeta gráfica y un ADSL potentes, pues visualizar videos en Second Life consume muchos recursos.

- <u>Visualizar nuestros videos y PDF</u>:

Es importante saber que si queremos ver nuestros vídeos o nuestros PDF en una televisión, es necesario ser propietario o tener derechos de propietario de la *Sim* donde pretendamos ver el video o PDF. Por lo tanto, si intentamos utilizar una televisión para ver nuestro video en el *Sim* en el cual no seamos propietarios, Second Life no nos permitirá visualizarlo.

- <u>Audio</u>: Tal y como se podía esperar de un mundo virtual que imita al mundo real, ha de contener sonidos. El problema fundamental es que actualmente son bastante limitados.

- Web: Linden Lab tiene su propio explorador para poder visualizar página webs.

- **Recreación de diferentes espacios.**

  Una de las características más diferenciadoras del mundo desarrollado por Linden Lab con respecto al mundo real es que si en el mundo real construyes en una tierra cualquier edificio y luego quieres cambiar cualquier aspecto del edificio o el edificio entero tardarías varios meses y sería muy costoso. Sin embargo, mediante una *Sim* de Second Life, cambiar el aspecto y las construcciones que tengamos hayamos hecho puede ser cuestión de horas.

- **Creación de avatares gratuitos.**

  A la hora de crear el avatar se puede modelar como nosotros queramos, haciéndolo alto, guapo, fuerte, con ojos azules, verdes, amarillos o con una mezcla de todos los colores.

  En Second Life se puede apreciar que todo el mundo es joven y atractivo. Éste ha sido considerado uno de sus mayores peligros, ya que como todo el mundo es joven puede haber menores de edad en Second Life que crean que la persona con la que están hablando es de su edad.

- **La moneda de Second Life.**

  SL tiene su propio dinero virtual, su moneda son los Linden dólar ( ). Los productos y servicios se compran y se venden utilizando esta moneda y, de esta manera, se da lugar a todo tipo de negocios en el mundo virtual. Por esta razón también el dólar Linden tiene un tipo de cambio oficial con el dólar de EE.UU.

- **La tierra.**

  Si se desea poseer tierra de Second Life es necesario pagar un impuesto "*tier*" a Linden Lab. Por esta razón, alternativamente se puede alquilar un terreno a otros jugadores en Second Life y así no pagar el impuesto "*tier*". En esta opción, hay que tener en cuenta que cada isla tiene sus propias normas y se deberían revisar antes de decidir si alquilar o no. Con cada alquiler tienes que tener claro cuantos *prims* puedes usar. Para más información acerca de las tierras se puede consultar la Wiki de Second Life [33].

- **Colaboración.**

  En SL es posible contactar sin límites geográficos con personas del mismo campo de interés de todo el mundo, compartiendo de este modo las dudas que se pueda tener en cualquier materia.

- **Interactividad y aprendizaje.**

  Para comunicarte en SL se puede utilizar la voz, el chat, videos, presentaciones, imágenes y gestor de tu avatar. De tal manera que Second Life dispone de todo el material necesario para que se pueda utilizar con fines educativos.

- **Buscar.**

  Second Life dispone de su propio buscador, que nos permite localizar cualquier tipo de cosa en SL.

- **Exposiciones.**

  Se pueden crear exposiciones de obras de artes como fotografías, cuadros, esculturas,… incluyendo información que describe a cada obra de arte de tal manera que se pueda aprender con sumo detalle sobre cada objeto expuesto en la galería.

- **Copias de material.**

  Se puede crear un objeto en Second Life y hacer copias ilimitadas, entregándolas a amigos o exponiéndolas para que cualquier persona que se acerque a la *Sim* pueda coger una copia.

  Esta es una característica de Second Life muy importante, pues en el mundo virtual hay muchísima gente dispuesta a ayudar que comparten sus objetos y están dispuestos a prestar ayuda e informar de manera muy amable.

  Esta capacidad de poder crear y hacer ilimitadas copias de los objetos de Second Life nos proporciona una importante herramienta para poder crear aulas educativas de forma gratuita.

Ejemplo:

Comentamos, por ejemplo, los siguientes objetos que podemos conseguir de manera gratuita en Second Life: casas, ropa, muebles, vehículos, cuadros, esculturas, manuales…

4.6. Utilidades que ofrece Second Life para la creación de entornos educativos

4.6.1. Campus virtual UC3M

Describimos a continuación cómo utilizar Second Life mediante Sloodle en el campus virtual de Moodle de la UC3M.

- Accedemos a la aplicación Aula Global 2 de la UC3M:

Figura 62. Aplicación Moodle Aula Global 2 de la UC3M

El administrador del sitio Web Moodle de la UC3M, utilizara este website de Moodle (campus virtual de la UC3M) para conectar las herramientas de Sloodle. De esta manera, tendremos conectada a dicha web la isla de la UC3M (Figura 4.11) con las herramientas Sloodle.

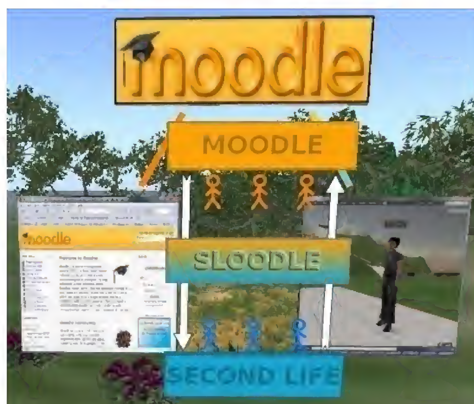


Figura 63. Isla de la UC3M en Second Life

4.6.2. Acceso a una isla para poder conectarse

Para poder acceder desde nuestro sito Moodle a una o varias islas, utilizaremos el siguiente enlace que tiene que insertarse en nuestros cursos de Moodle:



Vamos a explicar el proceso para poner un enlace de "teleport now" en uno de nuestros cursos. Es posible insertar varios enlaces en un curso y de esta manera cada enlace de "*teleport now*" nos conduzca a diferentes sitios de Second Life.

4.6.2.1. Subir la imagen "teleport now" a nuestro sitio Moodle

Vamos a subir la imagen "*teleport now*" a nuestro sitio Moodle para luego poder utilizarla de enlace en nuestros cursos Moodle.

- Entramos como administrador en nuestro sitio Moodle:

Nombre de usuario 100048464
Contraseña ••••••••• Entrar

- Seleccionamos el curso en el cual queremos poner los enlaces:



- Hacemos clic en el siguiente enlace:

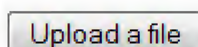


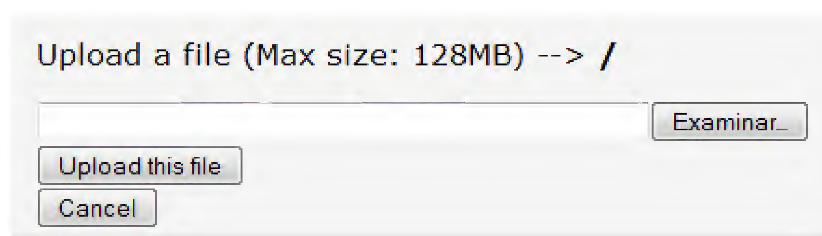
Este enlace previamente ha sido insertado en el curso para hacer referencia a la carpeta donde tenemos todo el material del mismo.

- Pulsamos el botón:



- Ahora pulsamos el botón *Upload a file* y en la ventana correspondientes seleccionamos la imagen a subir al servidor:

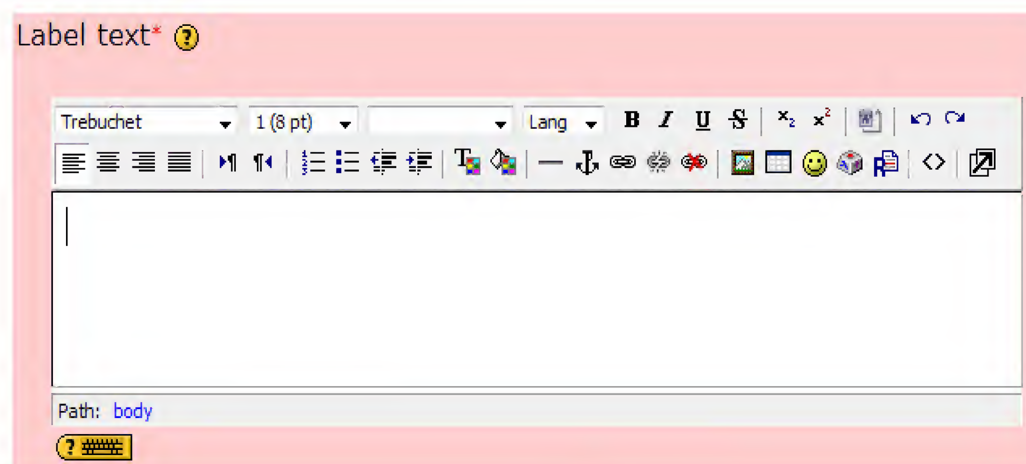




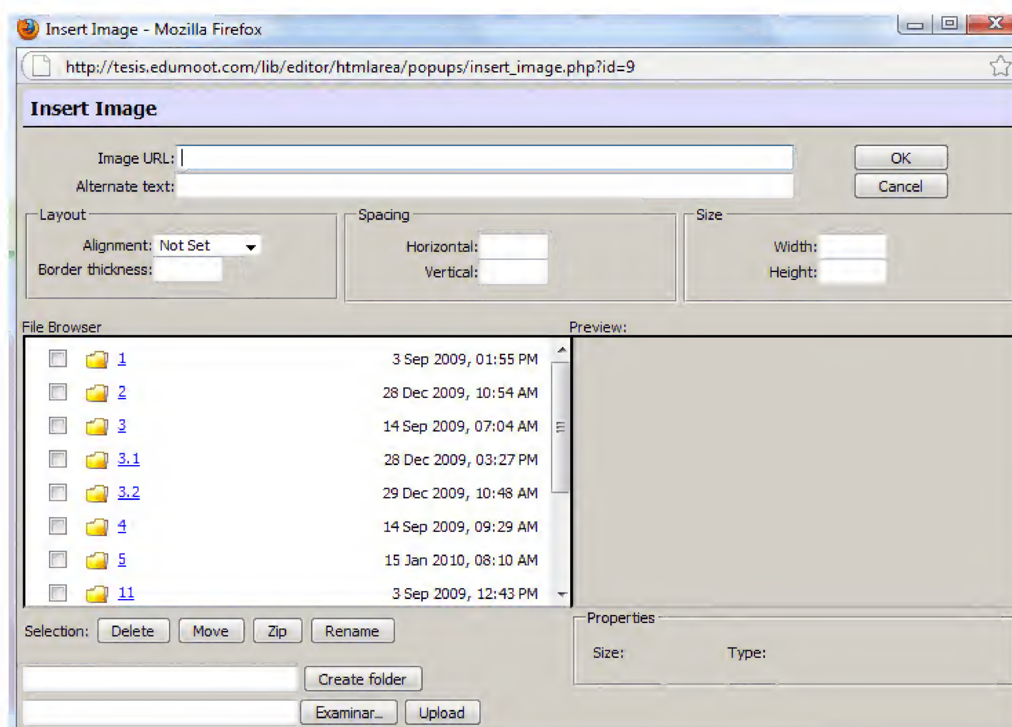
- Tras pulsar el botón *Upload this file*, comprobamos que la imagen se ha subido correctamente al servidor:



- En nuestro curso de Sloodle hacemos clic en el botón *Turn editing on*. Seguidamente accedemos al cuadro de texto *Add a resource* y seleccionamos la opción *Insert a label*. Con ello, visualizaremos la siguiente ventana:



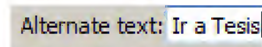
- Hacemos clic en el icono *Insertar Imagen* y seleccionamos la imagen teleport now en la pantalla correspondiente:



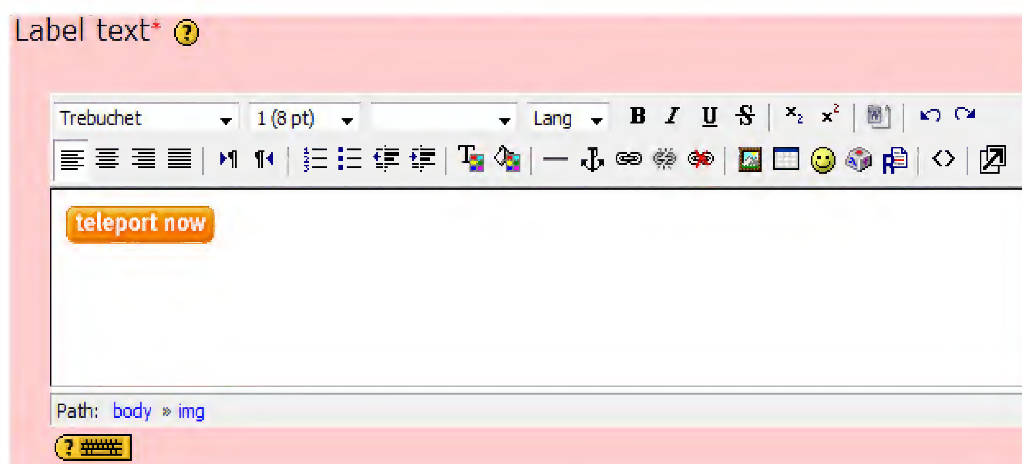
- Seleccionamos seguidamente la imagen:



- Insertamos un texto Alternativo:



- Observamos como en nuestra etiqueta se mostrará la imagen de "teleport now":

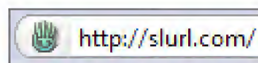


- Podemos centrar la imagen mediante el icono ≣.

4.6.2.2. *Búsqueda de la dirección SLurl*

Para buscar la dirección *SLurl* a la cual va a apuntar el enlace de nuestro sitio Moodle, debemos proceder con los siguientes pasos:

- Accedemos a la siguiente dirección web:



En esta web podemos visualizar el grid de Second Life (Figura 4.12), y haciendo clic en cualquier *Sim*, obtener su dirección.



Figura 64. Grid de Second Life (web SLurl)

- Hacemos clic en el enlace "*Build your own SLurls*", con lo que visualizaremos el formulario de búsqueda mostrado en la Figura 4.13.

SLurl
location-based linking in Second Life

Build your own SLurls!

Location data:

Region X Y Z

Options:

(leave these blank if not wanted)

Window title:

Window image: (full URL, ex: http://slurl.com/_img/default.gif)

Window message:

Generate SLurl Clear form

Your SLurl:

Go!

*This SLurlBuilder form was originally created by Ordinal Malaprop on 24 February 2006. Thanks Ordinal!

Figura 653. Búsqueda de direcciones SLurl

- En este caso introducimos en región la palabra *tesis* (correspondiente a la isla TESIS de la Universidad Politécnica de Madrid en Second Life):

Location data:

Region

tesis

- Hacemos clic en el botón "*Generate SLurl*":

Generate SLurl

- Vemos que nos aparece la siguiente dirección de Second Life:

Your SLurl:

http://slurl.com/secondlife/tesis//// Go!

- Pulsando el botón "*Go*", veremos cómo se ha localizado la isla TESIS (Figura 4.14).



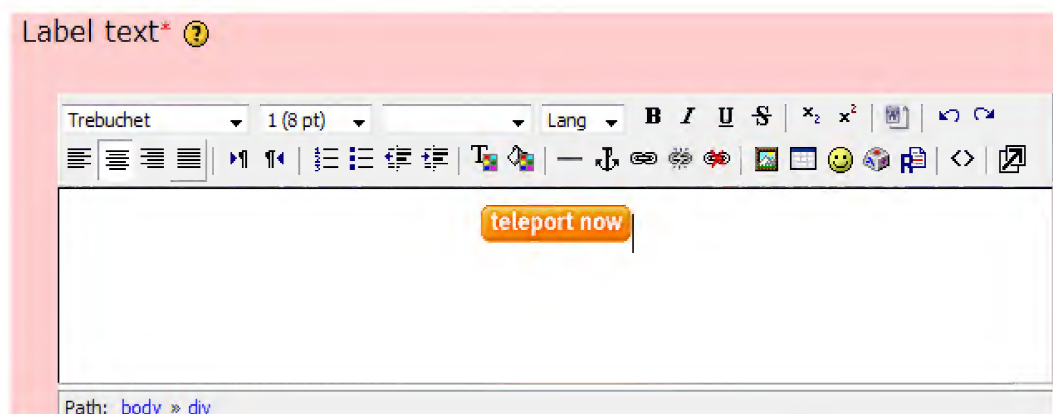
Figura 664. Localización de la isla TESIS en Second Life

Copiamos la dirección de la isla TESIS: http://slurl.com/secondlife/tesis////

4.6.2.3. Crear el enlace de nuestro sitio Moodle

Vamos a crear un enlace en nuestro sitio Moodle que apunte a la dirección del lugar de Second Life que seleccionemos.

- Accedemos a nuestro sitio Moodle:



- Seleccionamos la imagen *teleport now* y pulsamos el botón .

- Veremos la siguiente ventana:

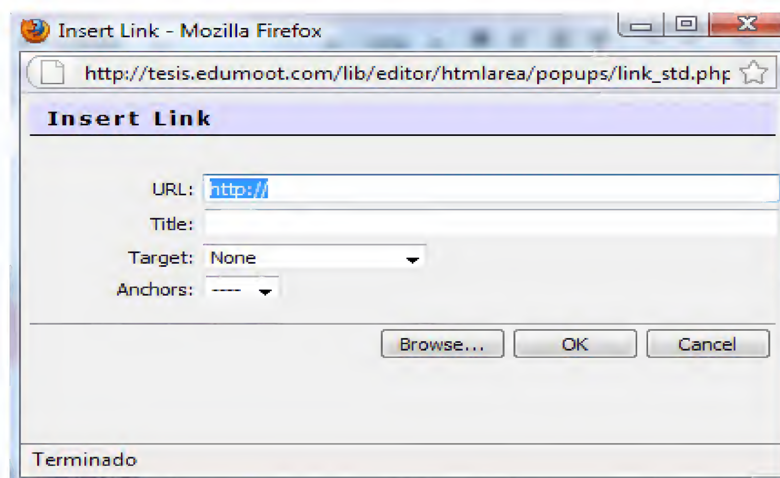


Figura 675. Inserción de enlaces en nuestro curso Moodle

- Insertamos la dirección de Second Life donde queramos que cuando hagamos clic en la imagen de "teleport now" acceda nuestro avatar:

URL: http://slurl.com/secondlife/tesis////

- Hacemos clic en el siguiente botón *OK* y regresamos al curso (Figura 4.16).



Figura 686. Visualización de nuestro curso Moodle con enlace a Second Life

4.6.2.4. Comprobar el enlace de Teletransporte

Vamos a comprobar el enlace de teletransporte.

- Hacemos clic en nuestro explorador de Second Life e introducimos nuestros datos de usuario:



Nombre: giaa
Apellido: Quixote
Contraseña: *********
Iniciar sesión
Recordar la contraseña

- Vemos que nuestro avatar está en la isla de Sloodle:



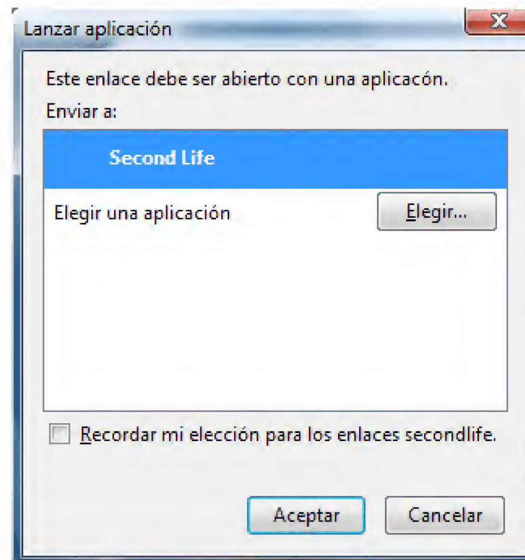
- En nuestro sitio Moodle, hacemos clic en la etiqueta **teleport now**:



- Nos conduce a la siguiente dirección Web:



- Hacemos clic en el enlace **teleport now** de esta página Web, con lo que visualizaremos la siguiente ventana:



- Tras pulsar el botón "*Aceptar*", vemos que en Second Life nos muestra el siguiente enlace:



- Pulsando el botón "*Teleportar*", vemos que somos conducidos a la isla TESIS:



De esta manera, cada curso puede contener varios enlaces de tipo "*teleport now*", para que cada tema del curso nos pueda conducir a diferentes islas de Second Life.

4.7. Problemas de Second Life

Enumeramos en esta sección los que pensamos que son los principales problemas que presenta Second Life.

- **Altos requerimientos de tarjeta gráfica.**

  Se trata de un entorno 3D y, por tanto, requiere de una tarjeta gráfica bastante potente, cuyas características dependerán del sistema operativo que tengamos instalado en nuestro ordenador. Los requisitos software y hardware que son requeridos para poder participar en Second Life pueden consultarse en la dirección web [60].

- **ADSL o Wireless.**

  Requiere de una buena conexión a Internet de banda ancha, pues toda la información de Second Life nos llega vía Web.

- **Sobrecarga de Sim en Second Life.**

  En las *Sim* en las cuales hay muchos avatares, se ralentizan los movimientos de nuestro avatar, hasta el punto que es posible que sea muy difícil controlarlo. El aspecto positivo es que podemos teletransportarnos a otra *Sim* que haya menos avatares para poder seguir disfrutando de Second Life.

- **Inestabilidad.**

  Second Life en cualquier momento puede tener una caída del sistema, lo que puede producir que se pierda nuestros amigos, sitios, la cuenta económica y objetos comprados (y sea imposible hacerles una copia).

Linden Lab no da ninguna solución para solucionar la perdida de objetos ni de dinero a causa de las caídas de su sistema. Muchos usuarios son reticentes a comprar objetos que no se puedan copiar, pues en cualquier momento se perder un objeto por el cual se haya pagado dinero y éste esté configurado para no poder hacer copias.

- **Actualizaciones de Second Life.**

  Es obligatorio actualizar periódicamente el explorador de Second Life, lo que nos puede afectar a la hora de:

  - Conectarnos a Second Life mediante consola:

    Nos conectamos a Second Life mediante consola para controlar un avatar mediante un programa informático, es decir, creamos un bot. En posteriores actualizaciones de Second Life pueden cambiar los protocolos utilizados, pudiendo esto causar que la librería que utilizamos para controlar el bot en Second Life quede desactualizada, lo que provocará que veamos el bot como una nube.

  - Conectarnos a Second Life mediante otro explorador distinto al que nos ofrece Linden Lab:

    Esto provocará también que al entrar en SL utilizando ese explorador veamos al avatar como una nube a causa de que hayan podido cambiar los protocolos de Second Life, y por supuesto el único navegador con el cual se podrá ver Second Life será el explorador de Linden Lab. Algunos ejemplos de exploradores para ver Second Life en otros mundos virtuales son Hippo [61] y Realxtend [62].

- **Chat.**

  La comunicación y funcionalidades del chat en grupos son paupérrimas.

- **La limitación de objetos por territorio.**

  Cada 512 metros de terreno pueden contener hasta 117 formas primitivas en su interior.

- **Herramientas para crear objetos.**

  Las herramientas de Second Life para crear cualquier objeto como puede ser una mesa, un libro, una silla,… son muy pobres a la hora de ofrecernos posibilidades para trabajar con ellas rápidamente, por lo que ala creación de objetos en Second Life puede llegar a ser un proceso tedioso. Aunque, también es cierto que se pueden utilizar herramientas de diseño que no son de Second Life, como *Poser* (Poser es un software de animación para los modelos que representan la figura humana en forma tridimensional; lo utilizamos para crear posturas y movimientos para nuestro avatar) y después subir nuestro trabajo a Second Life. El principal problema es no ser compatible 100%, es decir, lo que hagamos por ejemplo en *Poser* no se verá nunca exactamente igual cuando lo subamos a Second Life.

- **Pagar por tener tierras, por subir texturas, por subir imágenes, por subir videos,…**

  Como es de esperar Linden Lab tiene que conseguir beneficios al ser una empresa, y la forma de conseguirlos es cobrando por los servicios ofrecidos.

Solución:

Apostar por OsGrid, que es un mundo Virtual gratuito de código abierto, en el cual podremos tener tierras ilimitadas, podremos subir imágenes, vídeos… Por lo tanto, las universidades pueden contar con un DIAMANTE EN BRUTO, pues tienen la oportunidad de contar con un número ilimitado de aulas de forma rapidísima, y cada investigador puede tener su propio *Sim* para poder investigar en Second Life. El único recurso necesario para mantener OsGrid en una universidad será un host en el cual se almacenará toda la información de los objetos creados en este mundo virtual.

4.8. Ejemplos de entornos educativos en Second Life

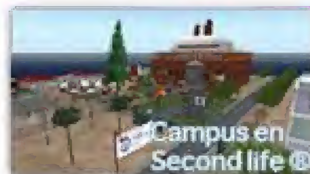
4.8.1. Second Life en la Universidad Carlos III de Madrid

Vamos a ver la *Sim* de la Universidad Carlos III de Madrid en Second Life, más adelante nos referiremos a ella como *Sim UC3M*.

- Accedemos a la dirección web de la biblioteca de la UC3M:

http://biblioteca.uc3m.es/

- Hacemos clic en el siguiente enlace:



- Visualizaremos la siguiente web:



Figura 697. Portal web de presentación de la UC3M en Second Life

- Los servicios ofrecidos que puede acceder cualquier usuario son:
    - Zona de trabajo de grupo.
    - Cursos de formación.
    - Información.
    - Actividades culturales.
- Hacemos clic en el siguiente enlace:

Transportarse al campus de la UC3M en Second Life®

- Visualizaremos entonces la web mostrada en la Figura 4.18.



Figura 708. Enlace de la localización de la UC3M en Second Life

- Hacemos clic en el enlace teleport now para teletransportarnos a la *Sim* de la UC3M en Second Life. Veremos entonces el panel mostrado en la Figura 4.19, dándonos la bienvenida e indicándonos las coordenadas de la UC3M en Second Life.

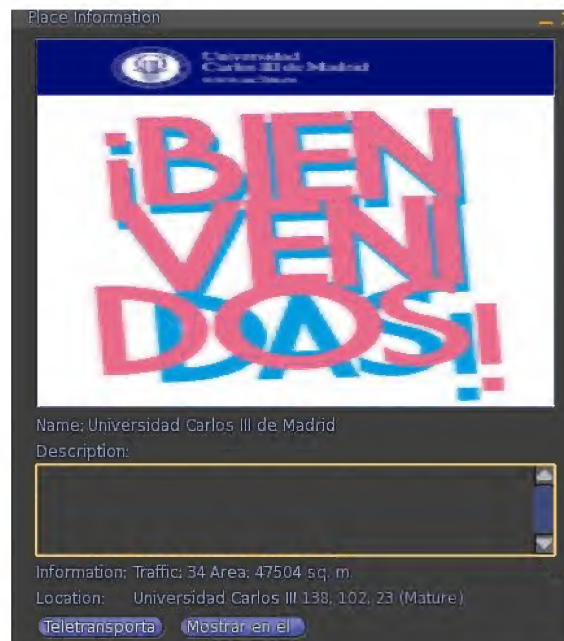


Figura 719. Panel de bienvenida a la Sim de la UC3M en Second Life

- Vemos el aspecto de la UC3M en Second Life (Figura 4.20).

  Podemos apreciar una réplica de la biblioteca de la UC3M, puesto que esta *Sim* esta mantenida y desarrollada por un antiguo trabajador de la biblioteca de la Universidad (Figura 4.21).



Figura 72. Sim de la UC3M en Second Life

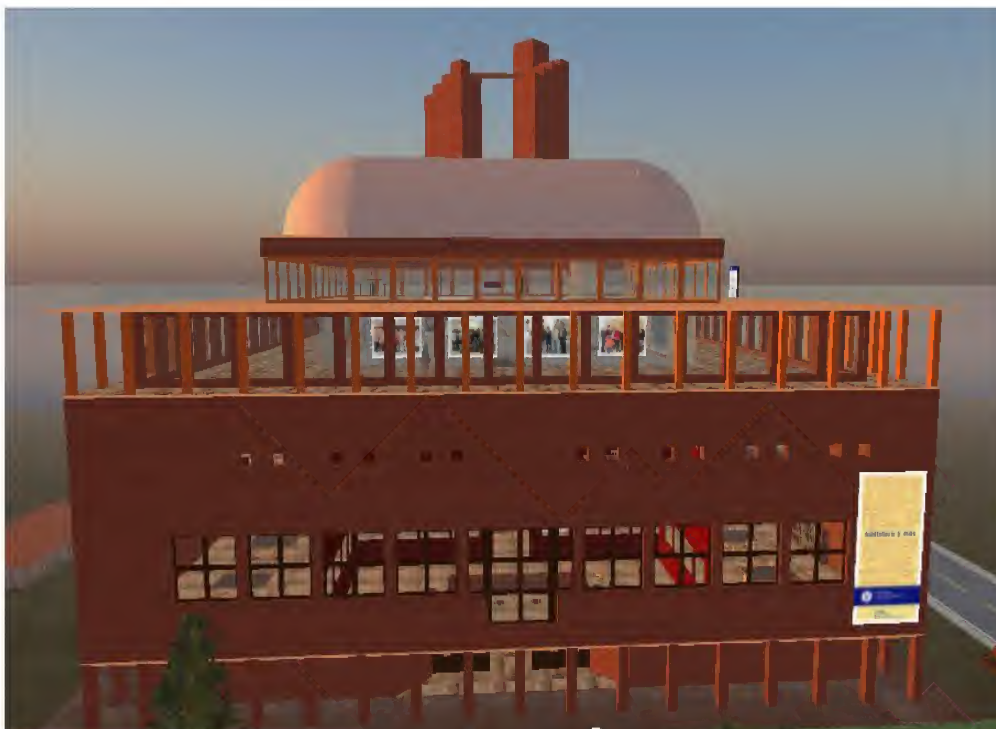


Figura 73. Aspecto exterior de la biblioteca de la UC3M en Second Life

- Podemos ver en la Figura 4.22 cómo la biblioteca virtual de la UC3M nos ofrece los mismos servicios y enlaces que la página web de la biblioteca UC3M



Figura 74. Interior de la biblioteca de la UC3M en Second Life

- Haciendo clic en el enlace de patentes, vemos cómo se nos informa de acceder a su respectiva página web (Figura 4.23).



Figura 75. Interior de la biblioteca de la UC3M en Second Life

- <u>Pulsando el botón "*Ir hacia pág*", accedemos directamente a la página web correspondiente (Figura 4.24)</u>.



Figura 76. Página web inicial de la sección de patentes en la biblioteca UC3M

4.8.2. Accesos a museos

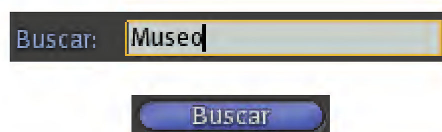
Describimos en esta sección como visitar un museo en Second Life.

- <u>Pulsamos el botón Buscar, visualizaremos la siguiente ventana (Figura 4.25), utilizada para buscar *Sim*, objetos, terrenos y demás lugares en el mundo virtual</u>.



Figura 77. Búsquedas en Second Life

- En el cuadro de texto de buscar, introducimos la palabra "Museo" y pulsamos el botón Buscar:



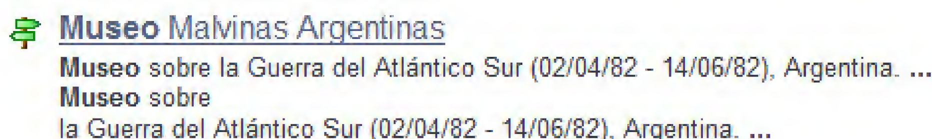
- La Figura 4.26 muestra los museos que se han localizado.



Figura 78. Resultado de una búsqueda en Second Life

Cada vez que hagamos una búsqueda en SL nos aparece un listado similar al mostrado en esta figura, al igual que sucede con los buscadores en internet.

- Hacemos clic en "Museo Malvinas Argentinas":



- Nos aparece la siguiente pantalla, con una descripción del enlace que hemos seleccionado:



Figura 79. Información proporcionada de cada ítem en una búsqueda

- <u>Pulsamos el botón “*Teleport*”, para visitar así el museo</u>.



Figura 80. Detalle de un museo en Second Life

- <u>Al investigar el museo, nos encontramos con un buen ejemplo de puntos a evitar</u>:
  - Sólo podemos ver fotos de baja calidad.

    En la red tenemos fotos de mucho mayor tamaño y con mejor calidad.
  - No se nos ofrece ninguna explicación sobre las imágenes que estamos viendo.
  - No aporta nada este sitio, nada más que parecer que estamos en un museo.
  - No existen más visitantes, estando.
  - Una isla donde no pasa mucha gente.

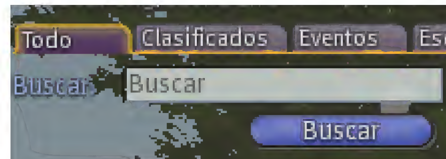
Una buena conclusión que debe extraerse de Second Life es que para desarrollar un buen proyecto educativo no es necesario emplear mucho dinero, lo que es realmente importante es OFRECER ALGO A LAS PERSONAS, y de esta manera tendremos visitantes en nuestros desarrollos. Desde el punto de vista educativo, ofrecer contenidos educativos a los estudiantes,

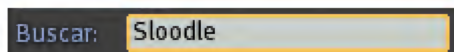
es algo más que poner a su disposición apuntes, guías, manuales… es esforzarse en explicar de forma muy clara y fácil, abundante información sobre varias materias.

4.8.3. Región Sloodle de Second Life

El *Sim* de Sloodle de Second Life, es un buen sitio para resolver dudas de forma online que puedan surgir sobre las herramientas de Sloodle, a través de los profesores de la isla.

- Pulsamos el botón Buscar e introducimos en la ventana correspondiente la palabra Sloodle:





- Como resultado de la búsqueda, hacemos clic en la región Sloodle:



- En la ventana correspondiente, seleccionamos la opción *Teleport* para trasladarnos a la isla:



- Con ello, ya estamos en la región Sloodle (Figura 4.29).



Figura 81. Región Sloodle en Second Life

- Debemos seguir el camino de piedras moradas para llegar a las aulas de Sloodle (Figura 4.30).



Figura 82. Aulas Sloodle

- <u>En la Figura 4.31 podemos observar un cartel que nos indica qué cursos hay disponibles actualmente sobre las herramientas de Sloodle</u>.



Figura 83. Aulas Sloodle

4.8.4. La Universidad de Sevilla

Analizamos en este apartado un entorno universitario desarrollado por la Universidad de Sevilla en Second Life (Figura 4.32). Realizando la búsqueda de manera similar a la descrita en los apartados anteriores, podemos localizarla fácilmente en el mundo virtual.



Figura 84. Universidad de Sevilla en Second Life

Accediendo a la Universidad, podemos darnos cuenta de la falta de paneles informativos que indiquen como movernos por ella, tal y como muestra la Figura 4.33.



Figura 85. Interior de la Universidad de Sevilla en Second Life

- Pulsamos el botón *"Minimapa"*, donde podemos observar que estamos solos en la universidad:



- Accedemos a la biblioteca disponible en la Universidad de Sevilla (Figura 4.34).



Figura 86. Interior de la Universidad de Sevilla en Second Life (I)

- Podemos observar cómo están organizadas las asignaturas (Figura 4.35).

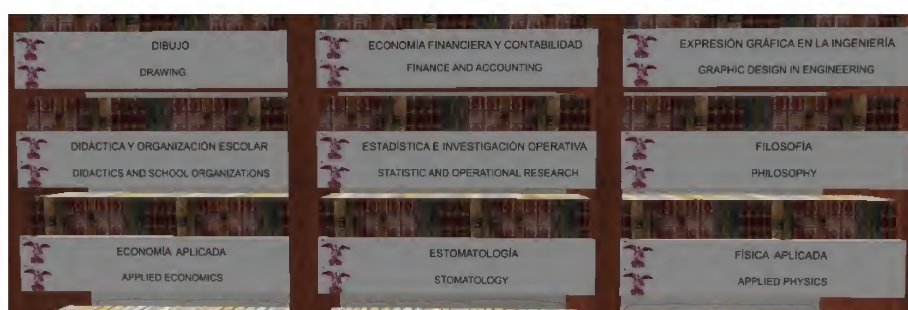
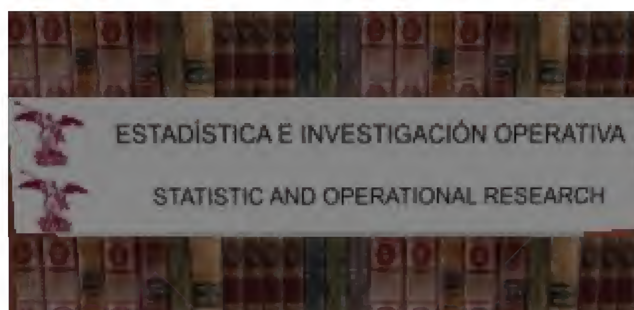
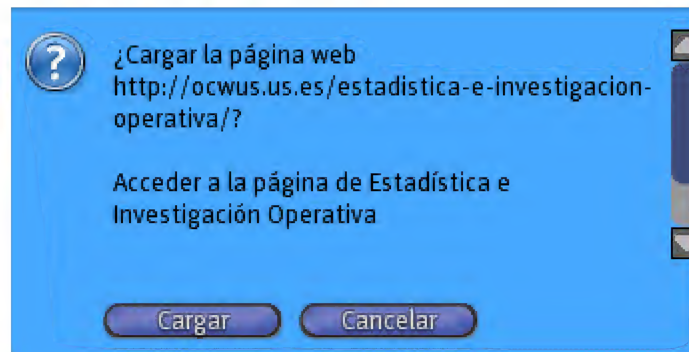


Figura 87. Interior de la Universidad de Sevilla en Second Life (II)

- Seleccionamos una asignatura y hacemos clic en los libros que observamos:



- Visualizamos un cuadro azul que nos indica si queremos acceder a la página Web de la asignatura:



- Pulsando el botón "*Cargar*", visualizamos la página web de la asignatura seleccionada (Figura 4.36).

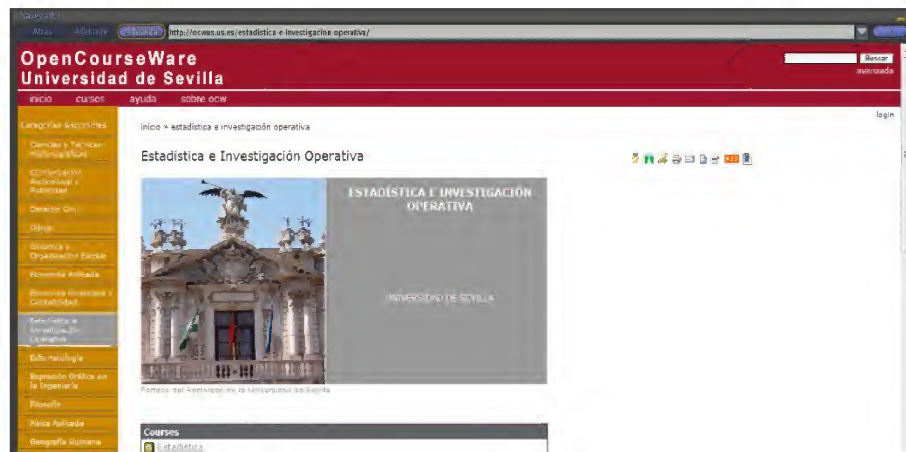


Figura 88. Acceso a asignaturas de la Universidad de Sevilla en Second Life

- Podemos observar en la biblioteca asientos para que los alumnos puedan conversar (Figura 4.37).



Figura 89. Interior de la biblioteca de la Universidad de Sevilla en Second Life

- Como en el mundo real, no podía faltar la cafetería de la Universidad (Figura 4.38).



Figura 90. Cafetería de la Universidad de Sevilla en Second Life

- Accedemos ahora a las aulas de docencia (Figura 4.39).



Figura 91. Aulas de la Universidad de Sevilla en Second Life

En las aulas no se ve observa ningún aparato de reproducción de video, de archivos PDF… para ayudar al profesor a dar las clases, y tampoco el profesor cuenta con suficiente espacio para llevar a cabo demostraciones prácticas dentro del aula.

También podemos observar es que sólo nos encontramos con dos aulas de docencia, por lo tanto podemos llegar a la conclusión que esta universidad virtual no está diseñada para impartir clases y disponer de contenidos adecuados, sino más bien, está diseñada como un entorno que está bien simplemente para estar un momento.

Como esta universidad virtual esta carente de cualquier material docente desarrollado para enseñar, se convierte en una estructura hueca, y por tanto vacía de personas.

4.9. Utilidades de Second Life

4.9.1. SCRATCH

Scratch es una utilidad que nos permite crear scripts en Linden Scripting Language (LSL), que es el lenguaje de Second Life.

4.9.1.1. Descargar y ejecutar SCRATCH

- Introducimos en el navegador web la siguiente dirección web, con lo que veremos la página web mostrada en la Figura 4.40.

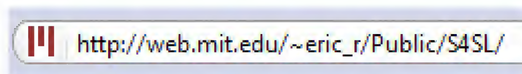
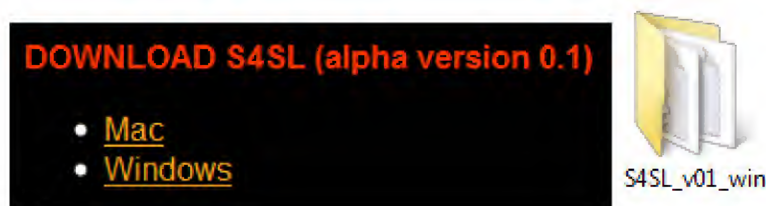




Figura 92. Página web de *SCRATCH*

- Hacemos clic en el enlace "Windows" para descargar S4SL y descomprimimos seguidamente el fichero descargado.



- Para ejecutar *SCRATCH*, accedemos a la carpeta *S4SL_v01_win* y simplemente hacemos doble clic con el botón izquierdo del ratón sobre el icono "*Scratch For Second Life*" (Figura 4.41):

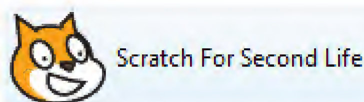




Figura 93. Utilidad *SCRATCH*

4.8.1.3. Crear scripts en SCRATCH

Vamos a crear nuestros primeros Scripts con SCRATCH.

- Vemos en la Figura 4.42 las pestañas de las que disponemos:



Figura 94. Utilidad *SCRATCH* (Panel)

Programaremos el código LSL utilizando estos botones.

- Hacemos clic en el botón “Control” y visualizamos las funciones de las que dispone este botón del menú (Figura 4.43).



Figura 95. Funciones incluidas en la opción Control de la utilidad *SCRATCH*

- Ahora, arrastramos el botón "*when I am touched*" a la zona de scripts:



- Hacemos clic en el botón "*Motion*" y vemos las funciones que nos ofrece (Figura 4.44):

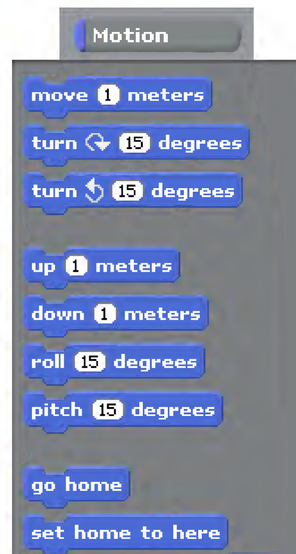


Figura 96. Funciones incluidas en la opción Motion de la Utilidad *SCRATCH*

- Arrastramos el botón "*turn 15 degrees*" a la zona de scripts:

Tal y como vemos en la Figura 4.45, acabamos de configurar que cuando se toque un objeto que contenga este código, entonces éste gire 15 grados hacia la derecha.



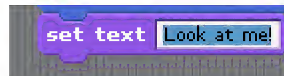
Figura 97. Configuración de eventos y acciones LSL con *SCRATCH*

- Introducimos una función más, para que pueda hablar (Figura 4.46):



Figura 98. Configuración de eventos y acciones LSL con *SCRATCH* (II)

- Hacemos doble clic en las letras de la función:



- Y escribimos el texto que queramos:



Con lo cual generamos que cuando se toque el objeto que contenga este código, entonces éste gire 15 grados hacia la derecha y diga TESIS (Figura 4.47).

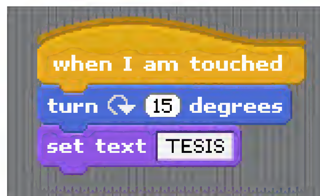


Figura 99. Configuración de eventos y acciones LSL con *SCRATCH* (III)

- Ya tenemos generado un script, ahora hacemos clic en el siguiente botón:



Sólo nos resta asignar a un objeto el código que acabamos de copiar.

4.8.1.4. Introducir un script en un objeto de Second Life

Accedemos a Second Life.

- Pulsando el botón derecho del ratón, nos aparecerá un menú con distintas opciones:



- Seleccionamos la opción *Crear* y veremos la pantalla mostrada en la Figura 4.48.



Figura 100. Opciones del menú Crear de Second Life

- Seleccionamos un objeto de este menú, en este caso, un cuadrado:



- Seleccionado el cuadrado, con el puntero del ratón elegimos una zona donde queramos crear el cuadrado, y haremos clic con el botón izquierdo del ratón:



- Hacemos clic con el botón derecho del ratón sobre el cuadrado y seleccionamos la opción *Editar*:



- Veremos entonces el menú mostrado en la Figura 4.49.

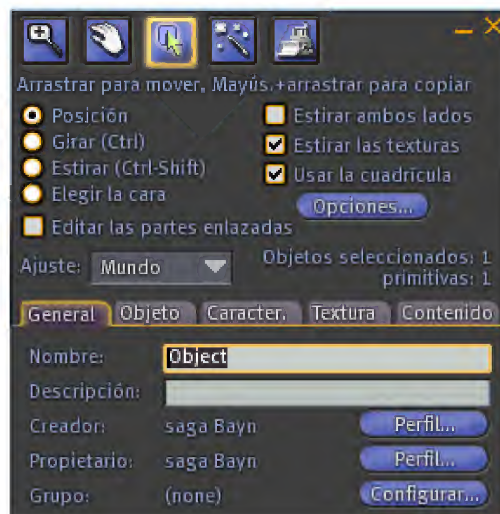


Figura 101. Opciones del menú Editar de Second Life

- Seleccionamos la pestaña Contenido y visualizamos la pantalla mostrada en la Figura 4.50.



Figura 102. Opciones de la pestaña Contenido en el menú Editar de Second Life

- Hacemos clic con el botón izquierdo del ratón en el botón "*Script nuevo*" y seleccionamos la opción "*New Script*":



- Visualizaremos la siguiente ventana:

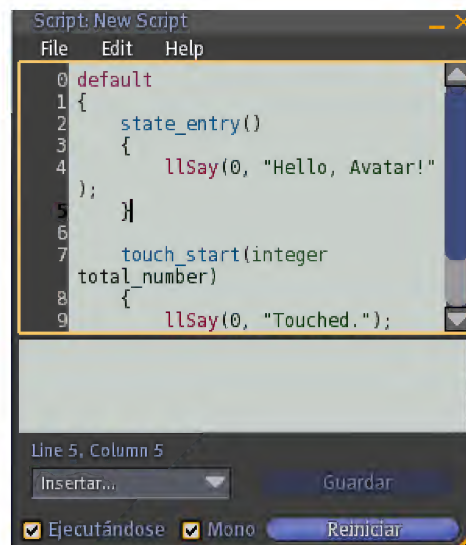


Figura 103. Asignación de código LSL a objetos

- Cambiamos el código que tiene dentro, por el nuevo código que hemos generado con la herramienta *Scratch*, seleccionando para ello la pestaña *Edit* y la opción *Paste* (Figura 4.52).

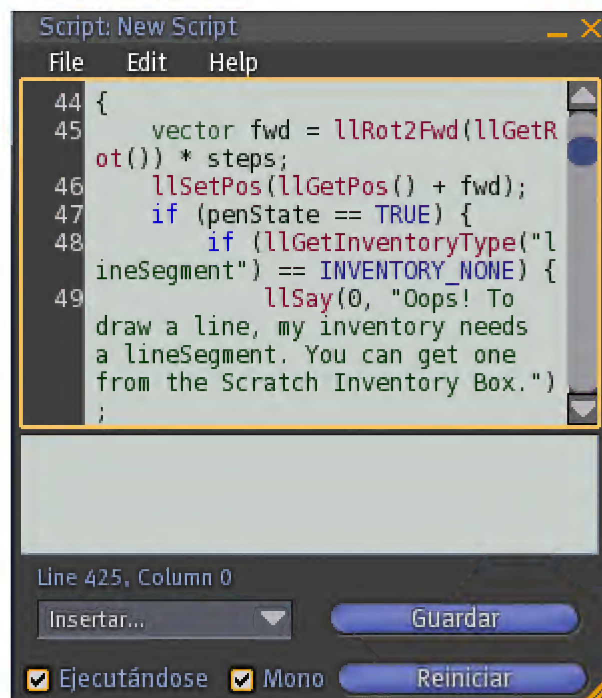
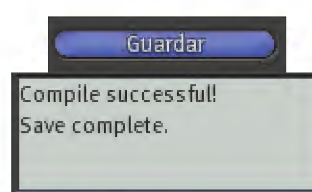


Figura 104. Asignación de código LSL a objetos (II)

- Pulsando el botón izquierdo del ratón en el botón *Guardar*, se nos informa que el código se ha compilado correctamente.



- Hacemos ahora clic en la caja y podremos observar cómo se mueve y muestra el mensaje asignado en el código:



Figura 105. Asignación de código LSL a objetos (III)

4.9.2. ROBO²

Robo² es un bot que acata las órdenes de uno o varios avatares que sean maestros de ese bot.

4.9.2.1. Descargar ROBO²

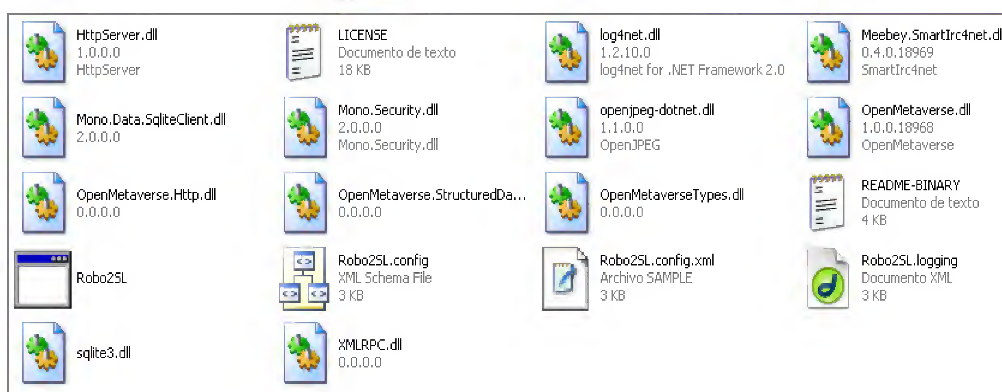
- Accedemos a la siguiente dirección Web:

  http://trac.robo2sl.org/display/robo/Releases+and+Source+Code

- Seleccionamos la última versión, y hacemos clic en *Binary ZIP*:



- Descargamos y descomprimir el fichero:



- De la carpeta *Robo2SL_Milestone_0905-1* nos interesan los siguientes dos ficheros:



En caso de no visualizar las extensiones de los ficheros, podemos configurar esta opción en Windows de la siguiente forma:

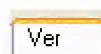
- Hacemos clic en *Herramientas → Opciones de carpeta*:



- Visualizamos la siguiente ventana:



- Seleccionamos la pestaña *Ver*:



- Observamos que está activada la opción de Ocultar las extensiones de archivo para tipos de archivo conocidos:



- Desactivamos dicha opción:



- Ahora podemos ver las extensiones de nuestros ficheros:



4.9.2.2. Configuración de ROBO^2

4.9.2.2.1. Conseguir el UUID del avatar máster del bot

En primer lugar, necesitamos el identificador UUID (identificador único universal), número que identifica a los objetos en Second Life) de un avatar para indicárselo al bot, y de esta manera, que éste sólo responda las órdenes de los avatares que tenga insertados como másteres.

- Nos conectamos a Second Life con el avatar "*Saga Bayn*":



- Accedemos a los ejemplos de la librería OpenMetaverse:

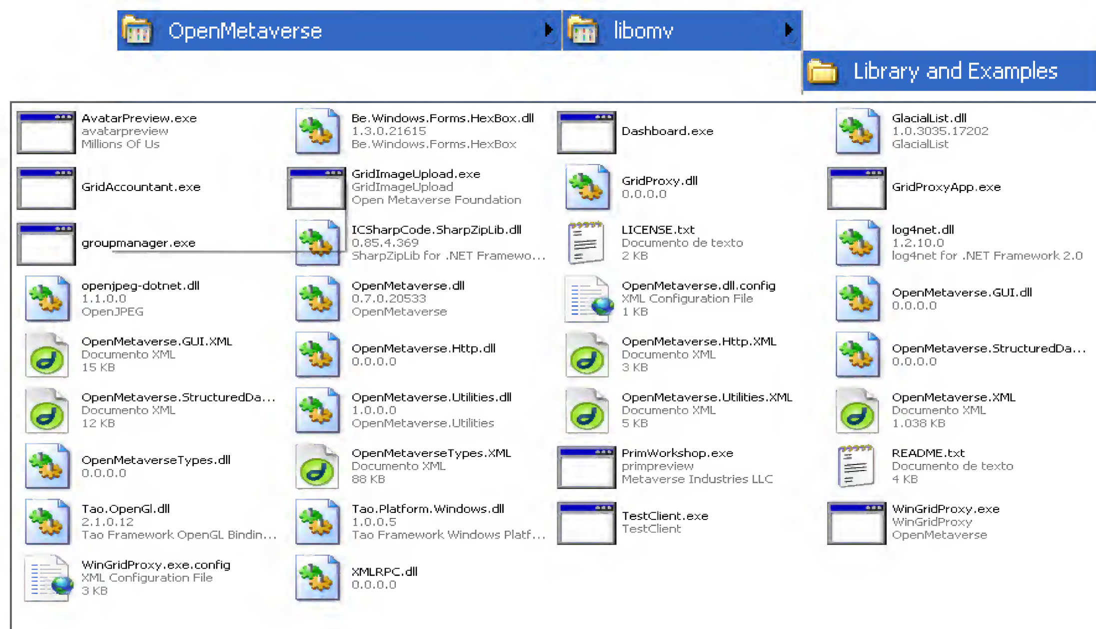


Figura 106. Ejemplos incluidos en la librería OpenMetaverse

- Hacemos clic en el fichero *GridAccountant.exe*, que nos mostrará los UUID de los avatares:



- Visualizamos la ventana mostrada en la Figura 4.55.

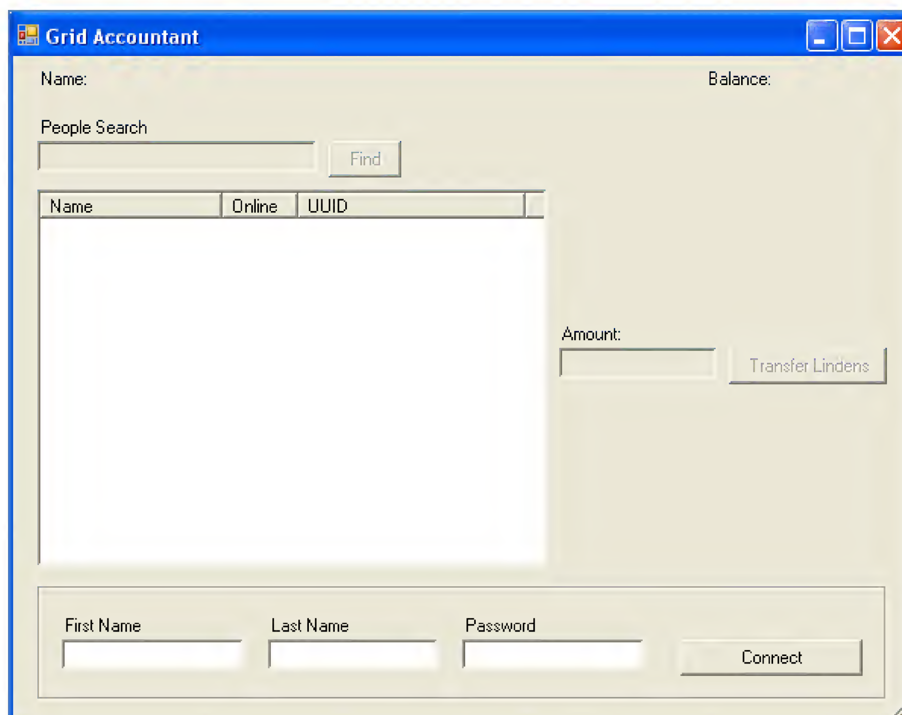


Figura 107. Búsqueda de los identificadores UUID en Second Life

- En la parte inferior, introducimos los datos del bot "*giaa Quixote*" y pulsamos el botón *Connect*:

Figura 108. Búsqueda de los identificadores UUID en Second Life (II)

- Desde el avatar "*saga Bayn*" vemos al bot "*giaa Quixote*" en Second Life:



- Introducimos en la parte superior dentro de la casilla *People Search* el nombre de "*saga Bayn*":

People Search
saga Bayn
Find

- Pulsamos el botón *Find* para que la aplicación nos facilite el identificador UUID del avatar "*Saga Bayn*":

Name	Online	UUID
saga Bayn	No	ff974b5b-1f07-4946-9adf-7e9782a24958

- Finalmente, pulsamos el botón *Disconnect*:



4.9.2.2.2. Configurar el fichero Robo2SL.config.xml.sample

En el fichero *Robo2SL.config.xml.sample* debemos introducir el nombre y apellido del bot, así como el UUID del avatar que queramos que sea maestro de ese bot.

- Accedemos a la carpeta "*Robo2SL_Milestone_0905-1*":



- Buscamos el fichero "*Robo2SL.config.xml.sample*":



- Editamos el fichero “*Robo2SL.config.xml.sample*”:

```
<!-- The 'Bots' section is an enumeration of BotInstances, we want to   -->
<!-- start when the program launches                                      -->
<Bots>
  <!-- if your bot hasn't seen its master before (don't know the UUID), -->
  <!-- you have to figure out the master's UUID first and provide it:   -->
  <Bot name="John Bot" password="PASSWORD"
       master="b9a7772e-912c-4990-dddd-e3a999993890" />
<?xml version="1.0" encoding="utf-8" ?>

<Robo2SL>

  <!-- These are the settings for the BotController (or global settings)  -->
  <!-- they must be embraced by the Controler node as shown here           -->
  <Controller>

    <!-- HTTPserver defines the settings for the built-in web server       -->
    <!-- port: default is 8080, so the URL would be http://server:8080/.. -->
    <!-- active: do we want the HTTP server to start at all?               -->
    <!-- useValidator: if 'false', the header checks are not applied,      -->
    <!--     which makes the server accessible from outside SL as well!    -->
    <!--     Only set that to 'false', if you apply other security checks! -->
    <HTTPServer port="8080" active="true" useValidator="true" />
  </Controller>

   <!-- the easiest notation afterwards is this one: (self-explanatory)  -->
   <Bot name="John Bot" password="PASSWORD" master="FORENAME SURNAME" />

   <!-- if you do have an IRC server you want the bot to connect to,     -->
   <!-- use this notation:                                               -->
   <Bot name="John Bot" password="PASSWORD" master="FORENAME SURNAME"
        ircServer="irc.server.example.com" ircPort="6667"
        ircChannelPrefix="john-" ircChannelPassword="blah"
        ircMaster="ircusername"/>
   <!-- please, set up your own IRC server, as the bot will spawn many   -->
   <!-- channels, and many IRC networks don't like that. also there can  -->
   <!-- be security issues if your channel by mistake does already exist -->
   <!-- or if the password is set too simple. -->


  </Bots>
</Robo2SL>
```

Figura 109. Fichero *Robo2SL.config.xml.sample*

- Borramos del fichero el siguiente código fuente:

```
<!-- the easiest notation afterwards is this one: (self-explanatory)  -->
<Bot name="John Bot" password="PASSWORD" master="FORENAME SURNAME" />

<!-- if you do have an IRC server you want the bot to connect to,     -->
<!-- use this notation:                                               -->
<Bot name="John Bot" password="PASSWORD" master="FORENAME SURNAME"
     ircServer="irc.server.example.com" ircPort="6667"
     ircChannelPrefix="john-" ircChannelPassword="blah"
     ircMaster="ircusername"/>
<!-- please, set up your own IRC server, as the bot will spawn many   -->
<!-- channels, and many IRC networks don't like that. also there can  -->
<!-- be security issues if your channel by mistake does already exist -->
<!-- or if the password is set too simple. -->
```

Figura 110. Fichero *Robo2SL.config.xml.sample* (II)

- El fichero queda de la siguiente manera:

```
<?xml version="1.0" encoding="utf-8" ?>

<Robo2SL>

  <!-- These are the settings for the BotController (or global settings)  -->
  <!-- they must be embraced by the Controler node as shown here          -->
  <Controller>
   <!-- The 'Bots' section is an enumeration of BotInstances, we want to   -->
   <!-- start when the program launches                                     -->
   <Bots>
     <!-- if your bot hasn't seen its master before (don't know the UUID), -->
     <!-- you have to figure out the master's UUID first and provide it:   -->
     <Bot name="John Bot" password="PASSWORD"
          master="b9a7772e-912c-4990-dddd-e3a999993890" />

   </Bots>
 </Robo2SL>
   <!-- HTTPserver defines the settings for the built-in web server      -->
   <!-- port: default is 8080, so the URL would be http://server:8080/.. -->
   <!-- active: do we want the HTTP server to start at all?              -->
   <!-- useValidator: if 'false', the header checks are not applied,     -->
   <!--     which makes the server accessible from outside SL as well!   -->
   <!--     Only set that to 'false', if you apply other security checks! -->
   <HTTPServer port="8080" active="true" useValidator="true" />
 </Controller>
```

Figura 111. Fichero *Robo2SL.config.xml.sample* (III)

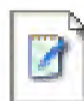
- Debemos modificar el siguiente fragmento de código:

```
<Bot name="John Bot" password="PASSWORD"
     master="b9a7772e-912c-4990-dddd-e3a999993890" />
```

- Introducimos, para ello, los siguientes datos:
  - Name: *giaa Quixote*.
  - Password: La contraseña del avatar *giaa Quixote*.
  - El máster del bot: El UUID del avatar *saga Bayn*.

```
<Bot name="giaa Quixote" password="granota09"
     master="ff974b5b-1f07-4946-9adf-7e9782a24958" />
```

- Modificamos la extensión del fichero "*Robo2SL.config.xml.sample*" eliminando la parte final "*.Sample*":


Robo2SL.config.xml.sample
Archivo SAMPLE
3 KB


Robo2SL.config.xml
Documento XML
2 KB

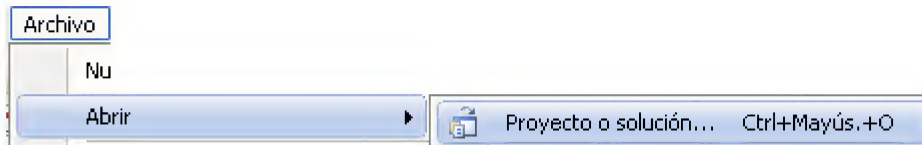
4.9.2.3. Ejecutar Robo² con Visual Studio

En este apartado vamos a arrancar Robo² mediante el programa Visual Studio para conectar así nuestro avatar en Second Life.

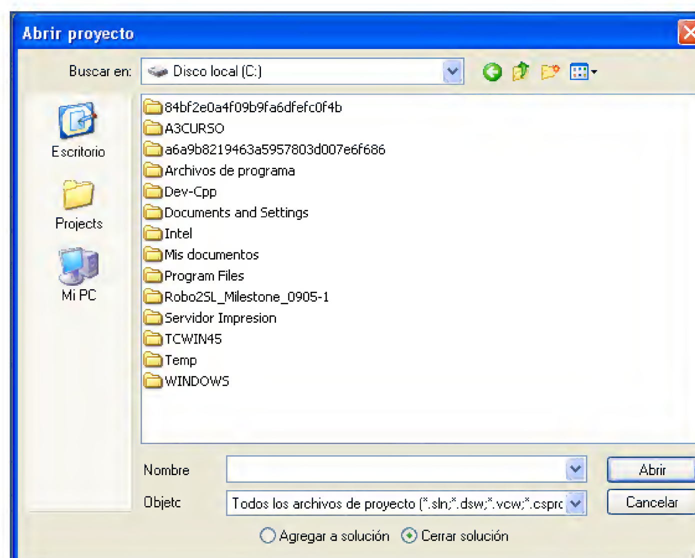
- Hacemos clic en nuestro icono de Visual Studio:



- Seleccionamos la opción *Archivo → Abrir → Proyecto o solución*:



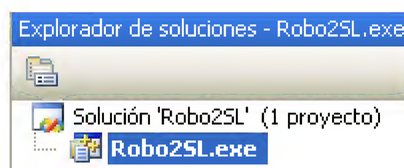
- Vemos la siguiente ventana para buscar un proyecto:



- Seleccionamos en esta ventana el fichero "*Robo2SL.exe*" de la carpeta "*Robo2SL_Milestone_0905-1*", y pulsamos el botón Abrir.



- Veremos entonces en el "*Explorador de soluciones*" el proyecto "*Robo2SL.exe*":



- Estando en "Visual Studio" presionamos la tecla "F5":

La ventana MS-DOS que visualizamos tiene que permanecer abierta mientras el bot esté en Second Life.

```
C:\Robo2SL_Milestone_0905-1\Robo2SL.exe
>> Robo² for Second Life, Open Source Framework        <<
>> Licensed under the terms and conditions of the GPL v2  <<
>> Copyright 2009 Jennifer Cole, Sienna M.             <<
>> For more details, please visit http://trac.robo2sl.org <<

>> Many thanks to                                      <<
>> * The Second Life Reverse Engineering Team (libopenmv) <<
>> * The project "Webserver in C#"                     <<
>> who published their code under a license which allows <<
>> us to use it within the Robo² project.              <<

Started. Unless the log4net logger is mis-configured, you
will not see much in the console, really. :-)
```

Figura 112. Ejecución de Robo^2 con Visual Studio

- Vemos que el bot "*giaa Quixote*" aparece en Second Life y nos remite el mensaje MI mostrado en la Figura 4.61.

El bot nos informa de su estado actual, en el cual no dispone de dinero y espera nuestras órdenes. Si queremos conocer la lista de comandos que podemos comunicarle, podemos introducir la palabra "*help*".

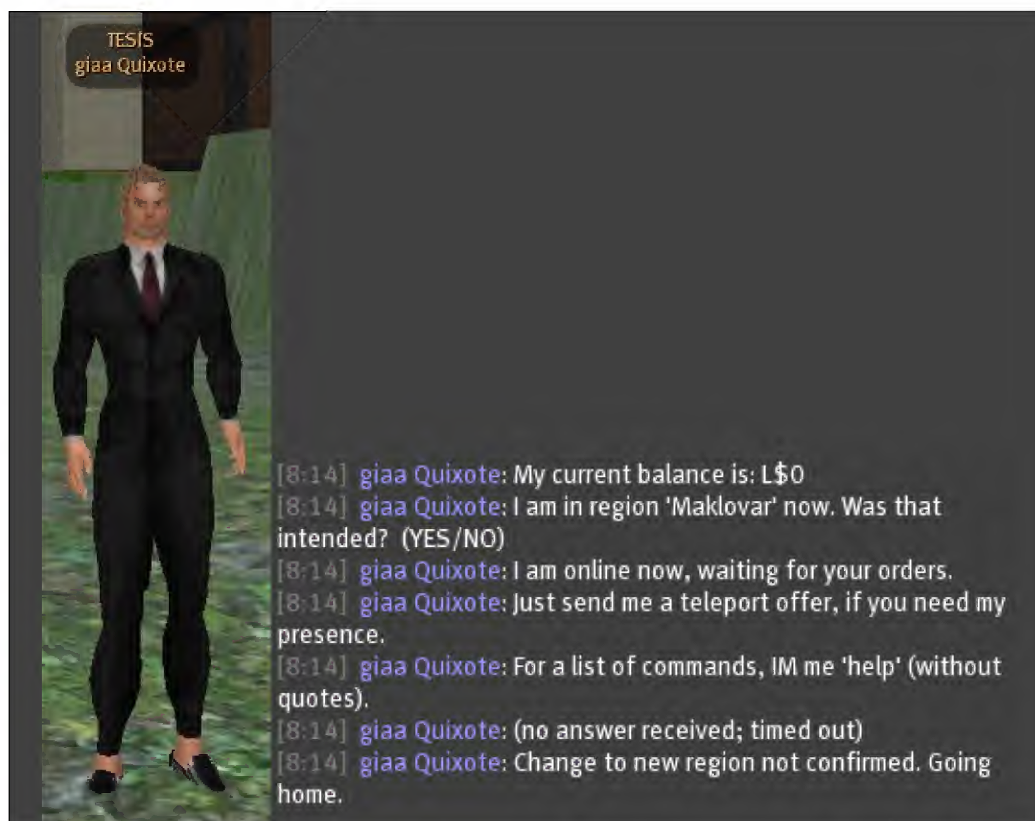
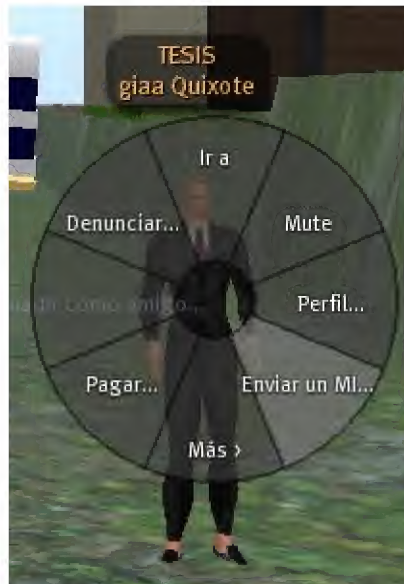


Figura 113. Estado del bot conectado a Robo^2

4.9.2.4. Comandos de Robo^2

En este apartado vamos a describir los diferentes comandos que posee Robo^2.

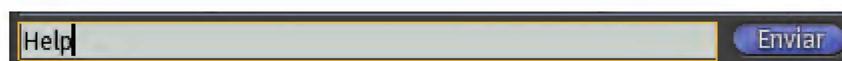
- Hacemos clic con el botón derecho del ratón sobre el bot, y seleccionamos la opción "*Enviar un MI…*":



- Accedemos con ello a la ventana que utilizaremos para controlar al bot "*giaa Quixote*":



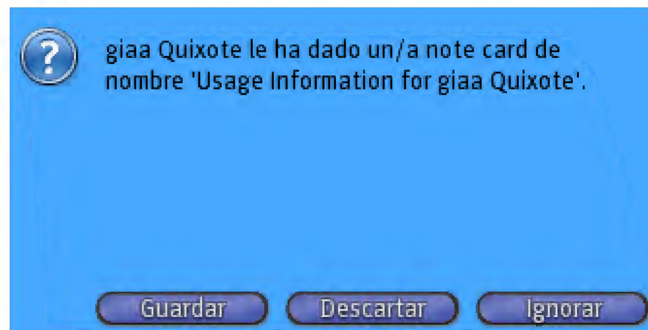
- Enviamos un MI a "*giaa Quixote*", le decimos "Help":



- Vemos que el bot "*giaa Quixote*" nos informa que nos ha remitido un mensaje.

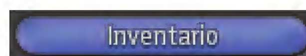
IM: giaa Quixote: Sending you a notecard... (depending on SL, it can take a moment)
IM: giaa Quixote: Notecard has been sent.

- Visualizaremos el siguiente menú:



- Pulsamos el botón "*Guardar*".

- Pulsamos a continuación el botón *Inventario, que esta en el explorador de Second Life*:



- Y vemos el mensaje que nos ha remitido el bot. Haciendo doble clic con el botón izquierdo del ratón sobre el mensaje, veremos la ventana mostrada en la Figura 4.62.

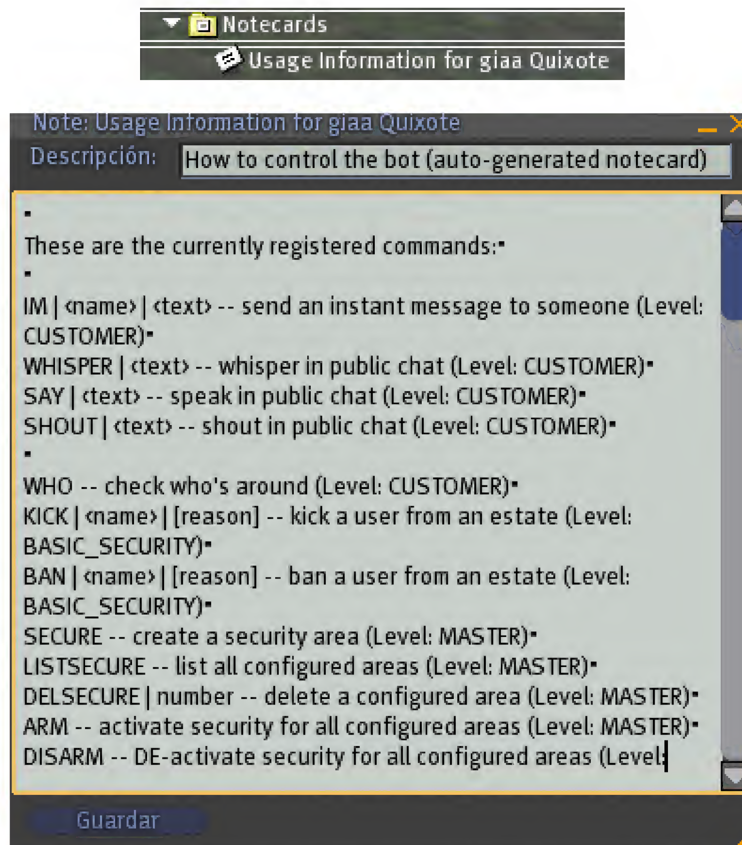


Figura 114. Mensaje con el conjunto de comandos de Robo[2]

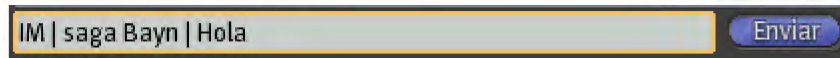
En este mensaje tenemos todos los comandos para controlar el bot.

a) Enviar un mensaje instantáneo a un avatar.

Utilizaremos para enviar un mensaje instantáneo a un avatar el siguiente comando:

IM | <name> | <text>

- Escribimos un MI a “*giaa Quixote*”, le decimos “Hola”:



- Pulsamos el botón “*Enviar*” y vemos como el bot “*giaa Quixote*” nos dice “Hola”.

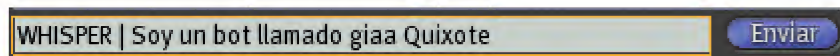
IM: giaa Quixote: Hola

b) Susurrar en el Chat público.

Utilizaremos para susurrar en el chat público el siguiente comando:

WHISPER | <text>

- Escribimos en el chat público “Soy un bot llamado giaa Quixote”:



- Vemos que “*giaa Quixote*” susurra el mensaje que le hemos transmitido:

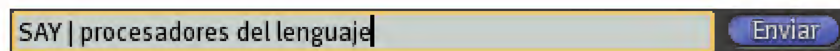
giaa Quixote susurra: Soy un bot llamado giaa Quixote

c) Hablar en el chat público.

Utilizaremos para hablar en el chat público el siguiente comando:

SAY | <text>

- Escribimos en el chat “Procesadores del Lenguaje”:



- Vemos que “giaa Quixote” dice el mensaje transmitido:

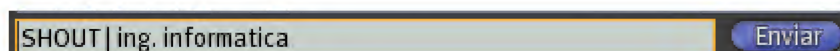
giaa Quixote: procesadores del lenguaje

d) Gritar en el chat público.

Utilizaremos para gritar en el chat público el siguiente comando:

SHOUT | <text>

- Escribimos en el chat público “Ing. informática”:



- Vemos que “giaa Quixote” grita “Ing. informática”:

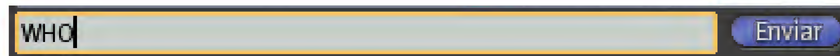
giaa Quixote grita: ing. informatica

e) Comprobar quién está alrededor.

Utilizaremos para ello en el chat público el siguiente comando:

WHO

- Escribimos en el chat público "WHO":



- El bot indica entonces las personas que están cerca:

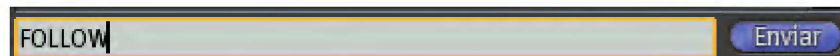
IM: giaa Quixote: These people are nearby: saga bayn, maryana velde, silva enimo, vox rossini

f) Permitir que el bot nos siga.

Utilizaremos para susurran en el chat público el siguiente comando:

FOLLOW

- Escribimos en el chat público "FOLLOW":



- El bot nos responde "*Affirmative!*", tras lo cual empezará a seguirnos:

IM: giaa Quixote: Affirmative!

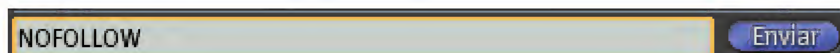
- Veremos como el bot nos sigue. Cabe destacar que no nos seguirá si nos teletransportamos).

g) Indicar al bot que deje de seguirnos.

Utilizaremos para ello en el chat público el siguiente comando:

NOFOLLOW

- Escribimos en el chat público "NOFOLLOW":



- El bot nos indica que nos espera donde está para futuras instrucciones:

IM: giaa Quixote: Waiting right here for further instructions.

h) Obtener la clave (UUID) de un avatar.

Utilizaremos para obtener la clave de un avatar el siguiente comando:

FINDKEY

- Escribimos en el chat público “FINDKEY” y el nombre del avatar del cual queremos obtener su UUID:



- El bot nos proporciona el UUID del avatar giaa Quixote:

IM: giaa Quixote: giaa Quixote's key is: 23ea8218-73dd-41a6-823c-218fec422bf2

i) Teletransportarse donde esté el bot.

Utilizaremos el siguiente comando para teletransportarnos a la posición del bot:

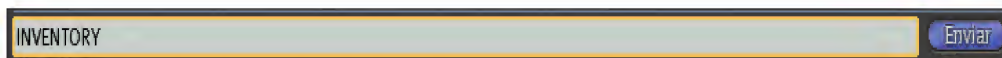
TAXI

j) Contenido del listado del inventario.

Utilizaremos para ver el contenido del listado del inventario el siguiente comando:

INVENTORY

- Escribimos en el chat público “INVENTORY”:



- El bot nos envía por el chat el contenido del listado del inventario:

```
[9:33] giaa Quixote: Edu (b1d3ed38-6ba0-0ac9-f461-5b034ad457e3)
 flight Feather (e635dad2-2690-c8d8-af4d-0bf8dc5b9baf)
 FREE HOUSES (108d8322-27bf-a419-4f0e-061df553adaf)
 FREE HOUSES (e71bb8e3-a592-fa98-da1b-e116671aec57)
 FREE LAND for the PEOPLE! Trileg, Boris (127, 128, 96) (bab190db-8d03-f107-6c8d-26894d4a5c0a)
 Free Sailboat! (e9890de1-be15-808b-9306-c69cba596566)
 FreeTube 3 (6b3220ee-8bc2-b32e-f0d4-fd21cf2e259a)
 FreeTube 3 (dcc125d1-c64d-c535-c9f3-1e1455fdb9bf)
 FreeTube 3 (6c0218e4-413e-d091-95a0-90405d015be7)
 FreeTube 3 (d02a0a15-d776-c089-c844-2093b0bf3b2b)
 Glasshouse/Greenhouse(copy) (8b48d24c-8c28-8a95-bbad-491bce0ccff9)
 How to play the wolverine claws (5fe925e0-6d20-8d0b-23ff-899080b85777)
 Hut Box (816520d1-ffaf-64ef-b330-5783f6393b2e)
 Hut Box 1 (c4ff915c-22b6-3c03-8478-d26cba10dde6)
 L'illa Diagonal Barcelona, Lilla Diagonal Barcelona1 (212, 123, (84852e85-789b-a8fc-0423-ec0cfcc36752)
 Last.fm DJ (d1c3b783-09d8-c852-3968-0f8d0b4ca22d)
[9:33] giaa Quixote: [ Tutorial ] ISLA N -AlterSpain (128, 128, 21) (73e62776-b4ab-3ea4-62d1-8d3f5ed8079d)
Calling Cards (9d5b7d64-422e-4e5c-8d70-2d97d0d43bad)
 mery Modan (f12da487-3b70-9203-7e9b-c4728b49eed7)
 saga Bayn (be042e78-d70e-81e8-cec7-3ae7947d62d0)
 Yeny Finl (41c28522-d0b0-4bcc-5f78-0a158444fb34)
Objects (c3c7b178-763e-49fd-85c4-14cbe7e5fa3f)
 SLOODLE 0.4 SET Revision 11 (890e2f6f-c54e-0616-4dd4-68ba044bf13c)
 SLOODLE Access Checker (f0238288-9bc9-262f-6349-ae99e988b93d)
 SLOODLE Access Checker Door (953fd177-8ed1-116e-9f83-81a1e4cacdbf)
 SLOODLE Choice (Horizontal) (138b2b57-3316-fd97-21bc-26aae5403751)
```

k) Obtener el valor LANDMARK.

Los LandMark son inventario que contiene ubicaciones precisas en Second Life. Su principal objetivo es permitir que la teletransportación a ese lugar de algún otro lugar.

Utilizaremos para que el bot nos dé un LandMark el siguiente comando:

LANDMARK

- Arrastramos un LandMark al bot “giaa Quixote”:

SLOODLE public sandbox, Sloodle (245, 2

- <u>Escribimos un MI a "*giaa Quixote*", le decimos "LANDMARK"</u>:

LANDMARK Enviar

- <u>El bot nos informa sobre el LANDMARK solicitado</u>:

giaa Quixote le ha dado un/a landmark de nombre 'SLOODLE public sandbox, Sloodle (245, 206, 22)'.

Guardar Descartar Ignorar

m) Teletransportar al bot donde estemos.

Vamos a ver cómo podemos teletransportar al bot "*giaa Quixote*" donde nosotros estemos, independientemente de que nos situemos en otra isla.

Nos encontramos en FREE LAND:

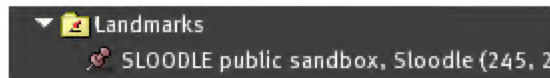


m.1) Teletransportar el avatar a Sloodle.

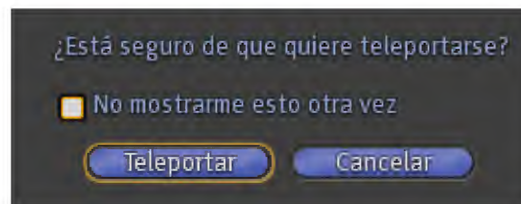
Vamos a posibilitar que el avatar "*saga Bayn*" se teletransporte a la isla Sloodle.

- <u>Con nuestro avatar pulsamos el botón *Inventario* y seleccionamos el LandMark de Sloodle</u>:





- Haciendo doble clic con el botón izquierdo del ratón en la LandMark, veremos la siguiente ventana:



- Pulsamos el botón *Teletransportar* y ya estaremos en Sloodle:



m.2) Ofrecer teletransporte al bot

Vamos a ofrecer teletransporte al bot para que venga así donde nosotros estamos.

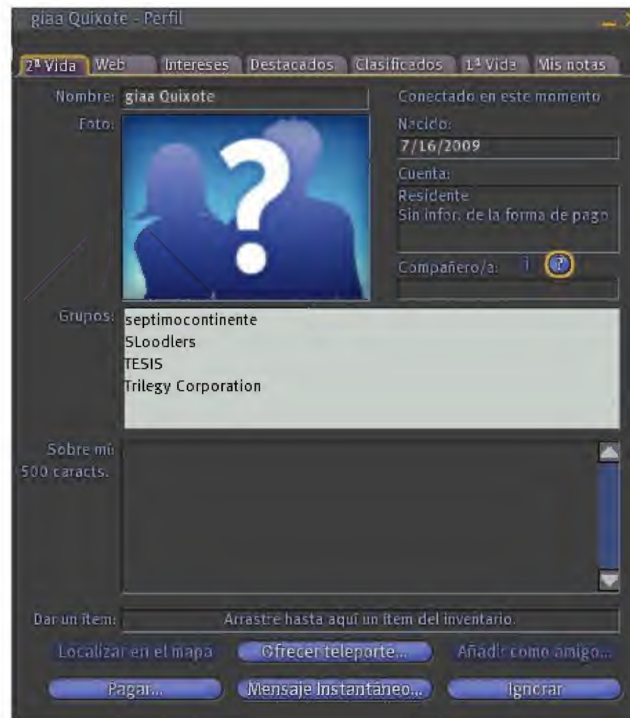
- Pulsamos el botón "*Comunicarse*":



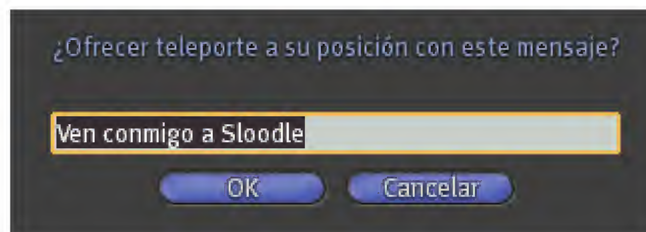
- Pulsamos entonces el botón "*Perfil*":



- Nos aparece la siguiente ventana:



- Hacemos clic en el botón "*Ofrecer teleporte…*", con lo que veremos la siguiente ventana:



- Pulsando al botón "*OK*", vemos como el bot "*giaa Quixote*" aparece:



m.3) Confirmar la opción de cambio de región.

Cuando aparece el bot "giaa Quixote" en la región Sloodle, nos comunica su llegada y nos pide que confirmamos la acción de haber cambiado de región.

[10:14] giaa Quixote: I am in region 'Sloodle' now. Was that intended? (YES/NO)

- Tenemos que responderle rápidamente "YES", sino regresará donde tenga fijada su casa en el mundo virtual:

[10:14] saga Bayn: YES

- Ahora el bot "giaa Quixote" añade esta región a su lista:

IM: giaa Quixote: Region added to whitelist.

Conclusiones Second Life

Second Life es un mundo virtual en 3D que ofrece en la modalidad de estudio e-learning la sociabilidad y la diversión, ya que mientras estudias puedes hablar y consultar dudas a tus compañeros y profesores mientras paseas por un entorno virtual.

El aspecto fundamental de esta nueva manera de enseñanza es la sociabilidad y diversión, ya que se hace amigos y además mientras estudias te diviertes.

La diversión es un aspecto muy importante en la enseñanza, ya que no hay que buscar tanto el fin de aprender como de disfrutar el viaje.

La enseñanza se vuelve divertida en Second Life, pues hay herramientas de Sloodle (Moodle + Second Life) que se basan en los programas de televisión (ruleta de la fortuna,…) para que los alumnos aprendan divirtiéndose y también se basan en juegos divertidos, como por ejemplo si fallas una pregunta te caes al agua.

Hay que destacar que esta manera de enseñanza social y divertida es totalmente innovadora y que solo se puede llevar a cabo mediante los mundos virtuales.

El mundo virtual de Second Life presenta la ventaja de ahorrar tiempo a los alumnos, ya que no tienen que desplazarse hasta el centro educativo, y posibilitar del mismo modo la asistencia y matriculación en cualquier centro educativo de cualquier parte del mundo.

CAPÍTULO V

SLOODLE

Capítulo 5

Sloodle

5.1. Introducción

Sloodle es una aplicación informática que permite la integración del software libre Moodle en el mundo virtual de Second Life**,** posibilitando el desarrollo de actividades educativas. De este modo, Moodle hace posible el desarrollo de cursos virtuales a medida, con espacios para impartir clases, disponer de ejercicios, foros, exámenes, etc. y SL posibilita el acceso virtual a ellos por medio de un avatar, participando en clases y conferencias en tiempo real. En síntesis, Sloodle es **Moodle + Second Life**.

Los principales objetivos de este capítulo son:

- Describir las principales funcionalidades de cada una de las herramientas de Sloodle.

- Facilitar una explicación sobre la configuración de cada una de estas herramientas.

- Proporcionar ejemplos de cada una de las herramientas, para mostrar así su funcionamiento.

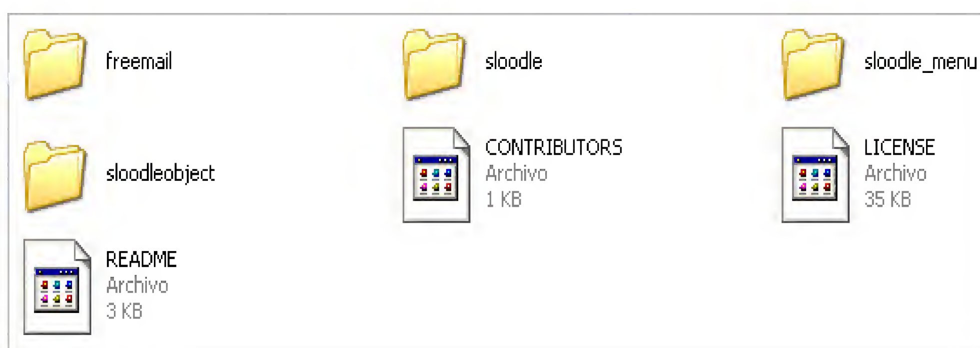
5.2. Instalación de Sloodle de forma local

Sloodle 0.4 es una aplicación *ALPHA*, así que no están incluidas todas las características completas, y ello puede provocar que no sea completamente estable y se deberían seleccionar versiones más estables para su instalación en servidores. Sloodle 0.4 es compatible con Moodle 1.8 y 1.9.

Para descargar Sloodle, hay que acceder a la página web:

http://www.sloodle.org

Descargado Sloodle 0.4, podemos encontrar en la carpeta correspondiente un fichero README, que nos informa sobre las funcionalidades e instalación de la aplicación.



5.3. Instalación del Módulo de Sloodle (requerido)

Vamos a instalar el módulo de Sloodle.

- Accedemos a la *mod* de Moodle:



MoodleWindowsInstaller-latest-19 / server / Moodle / mod

- Si existe en su interior una carpeta *sloodle*, es necesario borrarla para eliminar así instalaciones anteriores

- Añadimos la carpeta *sloodle* de esta descarga dentro de la carpeta *mod*, dado que Sloodle se configura para su instalación como un módulo adicional de Moodle.

5.4. Instalación de Block (opcional)

Esta aplicación se utiliza para poder visualizar en las páginas de Moodle un bloque con los detalles del avatar actual y con enlaces con funcionalidades importantes de Sloodle. Es recomendable para facilitar la navegación por las diferentes opciones que proporciona Sloodle.

Procedemos con la instalación de Block.

- Accedemos a la carpeta *blocks* de Moodle:



- Si existe en su interior una carpeta *sloodle_menu*, entonces borrarla para eliminar así instalaciones previas.

- Añadimos en su interior la carpeta *sloodle_menu* de está descarga:



5.5. Instalación de Sloodle Object Assignment plugin (opcional)

Se trata de un pluggin que nos permite utilizar una herramienta nueva llamada *Sloodle PrimDrop object*.

- Accedemos a la carpeta de Moodle *mod/assignment/type*:



- Si existe en su interior una carpeta *sloodleobject*, entonces borrarla para eliminar así instalaciones previas.

- Añade la carpeta *sloodleobject* desde esta descarga.

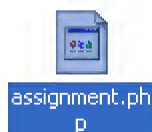


Necesitamos también modificar el paquete de lenguaje primario.

- Accedemos a la carpeta de Moodle *“lang/<langcode>”*:



- Dentro del directorio “*lang*” accedemos a la carpeta “*en_utf8*”, y localizamos el archivo “*assignment.php*”. Debemos añadir la siguiente línea inmediatamente antes del último “*?>*”:



$string['typesloodleobject'] = 'Sloodle Object';

```
$string['viewsubmissions'] = 'View $a submitted assignments';
$string['yoursubmission'] = 'Your submission';
$string['typesloodleobject'] = 'Sloodle Object';
?>
```

5.6. Instalación de FreeMail

- Accedemos a la carpeta de Moodle "*mod*".
- Si existe en su interior una carpeta *freemail*, entonces borrarla para eliminar así instalaciones previas.
- Añade la carpeta "*freemail*" de esta descarga.
- Hacemos click en el enlace "*Notifications*" para garantizar la configuración correcta del módulo.
- Visualizaremos con ello la pantalla mostrada en la Figura 5.1.

En esta figura vemos cómo se han creado en la base de datos de Moodle las tablas que necesita Sloodle. La razón es que los prims de Sloodle necesitan comunicarse con nuestro sitio Moodle mediante su base de datos.

Setting up module tables

UC3M ▶ Setting up module tables

No warnings - Scroll to the continue button

freemail

(mysql): CREATE TABLE `mdl_freemail` (`id` int(10) unsigned NOT NULL auto_increment, `course` int(10) unsigned NULL default '0', `name` varchar(255) NOT NULL default '', `value` varchar(255) NOT NULL default '', `description` text NOT NULL, `timemodified` int(10) unsigned NOT NULL default '0', PRIMARY KEY (`id`)) COMMENT='FreeMail'

Success

freemail tables have been set up correctly

sloodle

(mysql): SHOW TABLES

(mysql): SHOW TABLES

(mysql): CREATE TABLE mdl_sloodle (id BIGINT(10) unsigned NOT NULL auto_increment, course BIGINT(10) unsigned NOT NULL DEFAULT 0, type VARCHAR(50) NOT NULL DEFAULT '', name VARCHAR(255) NOT NULL DEFAULT '', intro MEDIUMTEXT NOT NULL, timecreated BIGINT(10) unsigned NOT NULL DEFAULT 0, timemodified BIGINT(10) unsigned NOT NULL DEFAULT 0, CONSTRAINT PRIMARY KEY (id))

Success

(mysql): ALTER TABLE mdl_sloodle COMMENT='Defines instances of Sloodle modules.'

Success

(mysql): CREATE INDEX mdl_sloo_cou_ix ON mdl_sloodle (course)

Success

sloodle tables have been set up correctly

Continue

Figura 115. Creación de las tablas de configuración de Sloodle en la base de datos de Moodle

- Tras pulsar el botón "*Continue*", se nos informa de la configuración correcta de las tablas de Sloodle.



Figura 116. Creación de las tablas de configuración de Sloodle en la base de datos de Moodle (II)

- Pulsando el botón "*Continue*", podemos ver la pantalla mostrada en la Figura 5.3:

The settings shown below were added during your last Moodle upgrade. Make any changes necessary to the defaults and then click the "Save changes" button at the bottom of this page.

New settings - SLOODLE Module

sloodle_allow_autoreg No Default: No
Allow auto-registration for this site

sloodle_allow_autoenrol No Default: No
Allow auto-enrolment for this site

Active object lifetime (days) sloodle_active_object_lifetime 7 Default: 7
The number of days before which an active object will expire if not used.

User object lifetime (days) sloodle_user_object_lifetime 21 Default: 21
The number of days before which a user-centric object (such as the Toolbar) will expire if not used.

Save Changes

Figura 117. Finalización de la configuración de Sloodle

En esta figura se muestra la configuración creada para Sloodle. Tras realizar los cambios que se requieran en los parámetros facilitados, simplemente resta pulsar el botón "*Save changes*" de esta página. Los parámetros que podemos configurar son:

- *Sloodle_allow_autoreg*: Indicamos si permitimos el autoregistro de nuestro portal o no.

- *Sloodle_allow_autoenrol*: Si permitimos que los usuarios se nos “autounan” a cada uno de los cursos.

- *Active object lifetime*: La cantidad de días que permitiremos que un objeto que hayamos creado en Second Life pueda mantenerse activo si no se usa.

- *User object lifetime*: La cantidad de días que un objeto de un usuario puede mantenerse activo si no lo usa.

5.7. Creación de un controlador

En este apartado resumimos cómo crear un controlador en Moodle.

- Pulsamos el botón *”Turn editing on”* en nuestro portal Moodle:

Turn editing on

- En el menu “*Blocks*” seleccionamos la opción *Sloodle Menu*:

Blocks

Add...

- Visualizaremos el siguiente menú Módulo, en el cual podemos comprobar la versión de nuestro módulo Sloodle instalada en Moodle (Figura 5.4).

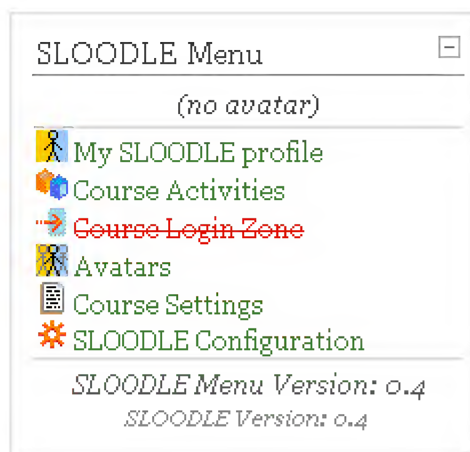
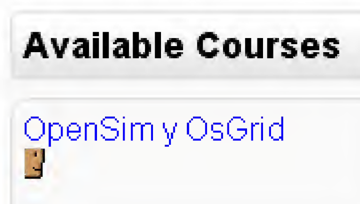
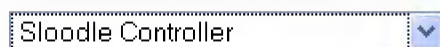


Figura 118. Menú Sloodle en Moodle

- Accedemos al curso que hemos creado en el capítulo anterior, *Curso de OpenSim y OsGrid*:



- Pulsamos sobre en la actividad del Tema 1 en *Sloodle controler*:



- Le asignamos el nombre "*Controlador Descargas y Carpetas*" e insertamos una contraseña:

Agregando Sloodle Module a tema 1

Ajustes generales

Module Type Sloodle Controller

Name* Controlador chat Descargas y Carpetas

Descripción*

Trebuchet 1 (8 pt) Idioma

Controlador para el Chat de Descargas y Carpetas

Sloodle Controller

Enabled You can control access to your courses by enabling or disabling the Sloodle Controllers

Prim Password 123456

Figura 119. Creación de un controlador en Sloodle (I)

- Acabamos de crear en el Tema 1 el siguiente controlador:



- Hacemos clic en el "*Controlador Chat Descargas y Carpetas*", comprobando que nos aparece como controlador habilitado.

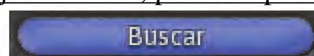
Status: Enabled

Figura 120. Creación de un controlador en Sloodle (II)

5.8. Creación del Podio

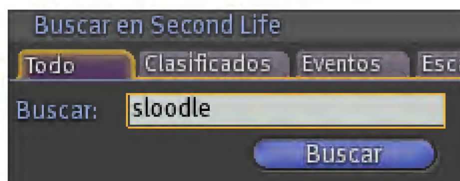
En este apartado vamos a conectarnos a Second Life para configurar las herramientas de Sloodle, las cuales nos va a permitir "rezear[2]" los distintos objetos Sloodle.

- Vamos a seleccionar el objeto *Sloodle*, para ello pulsamos el botón *Buscar*:



- Veremos un cuadro de dialogo, y en la pestaña *Todo*, ponemos en la opción de *Buscar* la palabra "Sloodle", y pulsamos el botón "Buscar":

[2] Crear un objeto en un simulador de Second Life, bien sea desde el inventario o desde la herramienta de construcción.



- En el cuadro de dialogo que veremos, hacemos clic en el enlace *Región: Sloodle*:

Región: Sloodle

- Seguidamente, pulsamos el botón *Teleport*:

Teleport

- Vamos enlaces a las distintas partes de la isla de Sloodle:

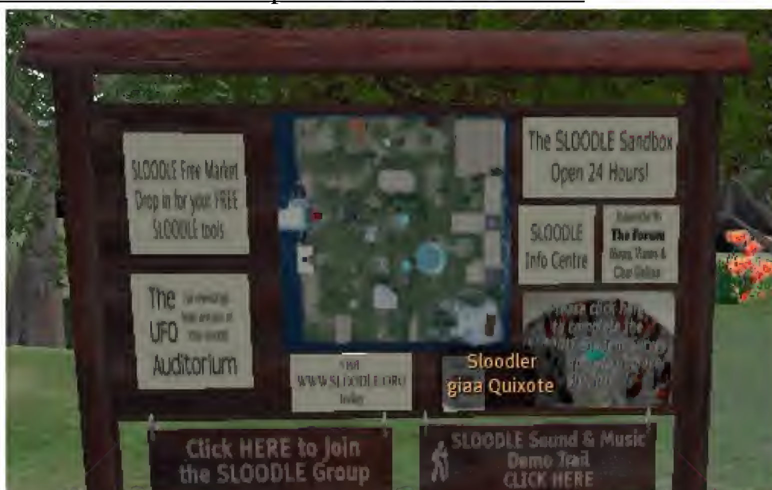


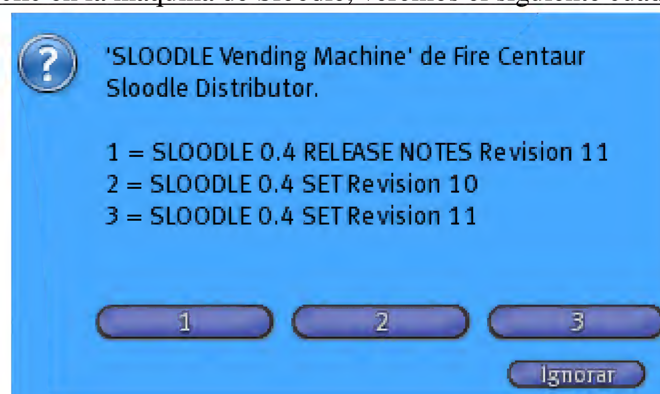
Figura 121. Enlaces disponibles en la isla Sloodle

- Los enlaces disponibles, hacemos clic en *Sloodle Free Market*, para conseguir las *Sloodle tols* en la máquina que visualizaremos (Figura 5.8).

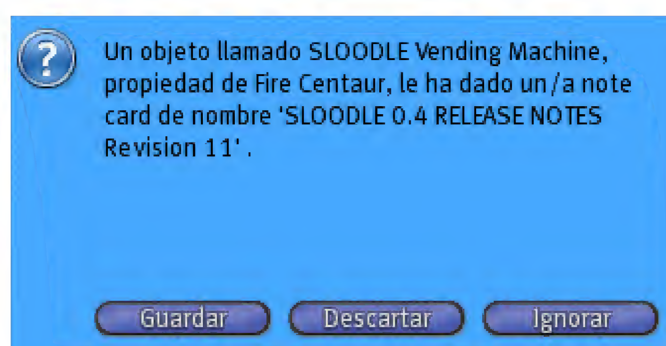


Figura 122. Máquina para descargar las Sloodle tools

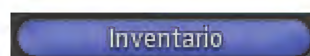
- Tras hacer clic en la máquina de Sloodle, veremos el siguiente cuadro:



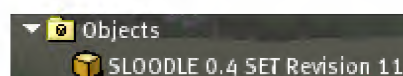
- Elegimos el botón número "*3*", y seleccionamos la opción *Guardar* en el cuadro correspondiente:



- Accedemos ahora a nuestro inventario:

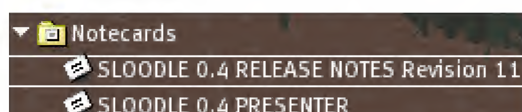


- Veremos que en la carpeta *objects*, aparece el objeto descargado *Sloodle 0.4 SET Revision 11*:



Podemos observar cómo coincide la versión del objeto Sloodle (0.4) con la versión de Sloodle de nuestra web Moodle.

- Repetimos el proceso, haciendo de nuevo click en la máquina de Sloodle y seleccionamos ahora botón número "*1*". Tras seleccionar la opción Guardar y acceder de nuevo a nuestro inventario, veremos que el objeto descargado aparece en la carpeta en la carpeta *Notecards*:



- Vamos a la zona de *Sandbox* de sloodle y arrastramos el objeto *Sloodle 0.4 SET Revision 11* al suelo:



- Lo tocamos para configurarlo:



Figura 123. Podio de Sloodle

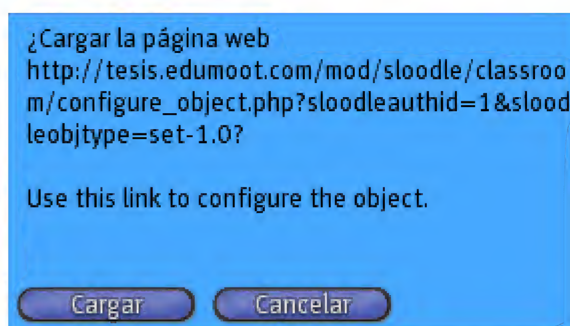
- Nos informa que insertemos la dirección de nuestro sitio Moodle en el Chat:



- Introducimos en el Chat la dirección de nuestro sitio Moodle:

  http://tesis.edumoot.com

- Veremos el siguiente mensaje:



- Tras pulsar el botón *Cargar*, se nos solicita regresar a la página Web de nuestro Moodle, e introducimos nuestros datos de administrador:

Returning to this web site?

Login here using your username and password (Cookies must be enabled in your browser)

Username

Password

Login

- Veremos la siguiente ventana:

Home ▸ SLOODLE Module ▸ SLOODLE Object Authorization

Object Details

Object Name: SLOODLE 0.4 SET Revision 11

Object UUID: 7199c7a7-9d1e-b042-4145-d78853c7a3eb

Object Type: set-1.0

SLOODLE Object Authorization

Select Controller:

3. OpenSim y OsGrid

SLOODLE Controller: Controlador descargas y carpetas

Do you want to authorize this object? Yes

Submit

- Hacemos clic en el "controlador nuevo" elegido:

3. OpenSim y OsGrid

SLOODLE Controller: Controlador descargas y carpetas

- Pulsamos el botón *Submit* y visualizaremos la siguiente ventana:

Home ▸ **SLOODLE Module** ▸ **Object Configuration**

Authorizing for: OpenSim y OsGrid > SLOODLE Controller: Controlador descargas y carpetas

Object authorization has been successful.

Object Details

Object Name: SLOODLE 0.4 SET Revision 11

Object UUID: 7199c7a7-9d1e-b042-4145-d78853c7a3eb

Object Type: set-1.0

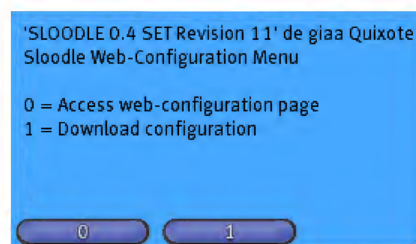
Object Configuration

No additional configuration options for this object.

- Accedemos de nuevo al podio:



- Hacemos clic sobre el botón "**1**":



- Ahora nos fijamos de nuevo en el podio:



El podio nos indica que el objeto se ha configurado correctamente y está listo para ser usado.

5.9. Access Ckecker

5.9.1. Definición de Access Ckecker

Esta utilidad permite realizar dos funciones principales al entrar el avatar en su espacio:

- Si el avatar no está registrado como usuario en la plataforma, Moodle lo obligará a su registro.
- Si el Avatar está registrado en un curso, se le dará la bienvenida.

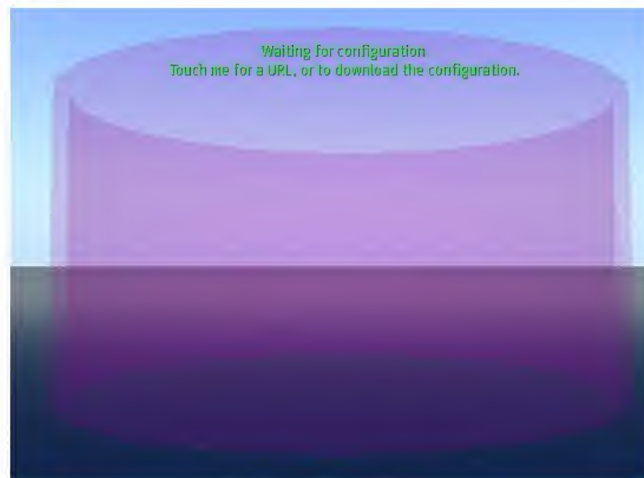


Figura 124. Utilidad Access Checker

5.9.2. Configuración de Access Checker

Vamos a configurar Access Ckecker.

- <u>Hacemos clic sobre la funcionalidad "*Rez objects*" (crear objectos)</u>:



- Nos muestra el siguiente menú:

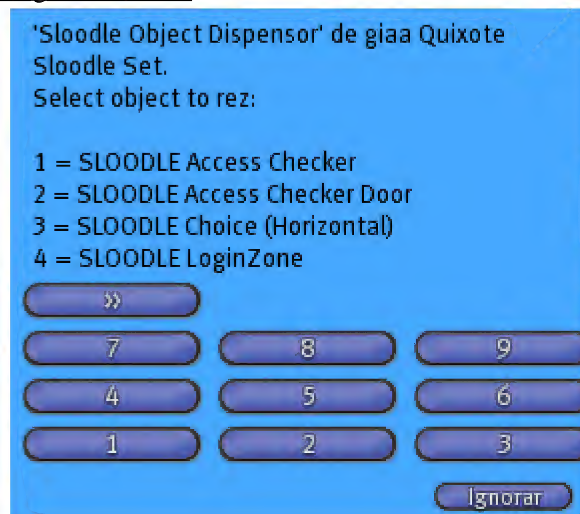


Figura 125. Configuración de Access Checker (I)

- Pulsando el botón "*1*" visualizaremos *Access Ckecker* (contorno rosa que se observa en la Figura 5.12).

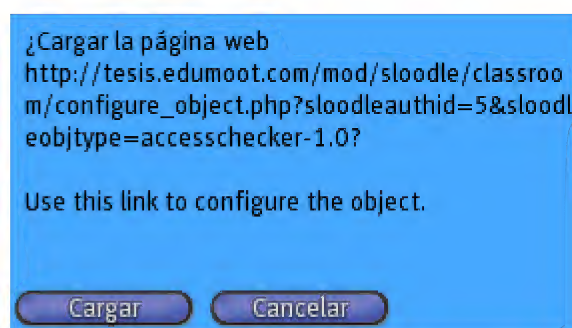


Figura 126. Configuración de Access Checker (II)

- Hacemos clic sobre “Access Ckecker” para introducirle la URL, pues está esperando la configuración:



- Nos muestra el siguiente cuadro:



- Hacemos clic en el botón “*Cargar”* y se nos muestra la ventana mostrada en la Figura 5.13.

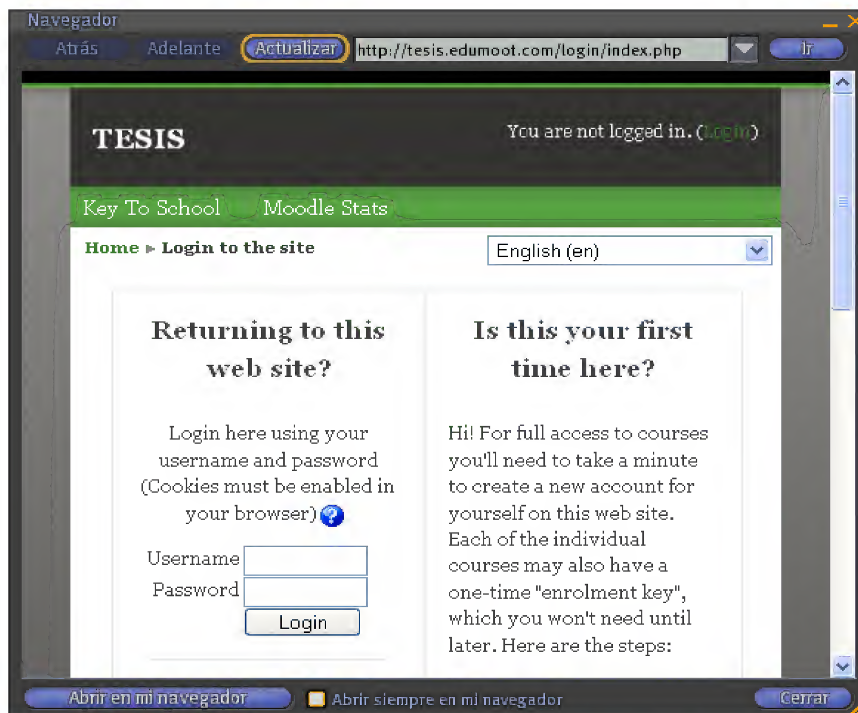


Figura 127. Configuración de Access Checker (III)

Introducimos nuestro nombre de usuario y contraseña.

- Nos muestra la siguiente ventana:

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE Access Checker

Object UUID: 2dcc346e-df44-0bea-3d1f-3c8a7f6e2cde

Object Type: accesschecker-1.0

Object Configuration

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Submit

Figura 128. Configuración de Access Checker (IV)

- Pulsamos el botón *Submit* y visualizaremos la pantalla mostrada en la Figura 5.15.



Figura 1295. Configuración de Access Checker (V)

- Pulsamos el botón *Continuar* y hacemos de nuevo click sobre el sombreado rosa de *Access Ckecker*. Con ello, visualizaremos el cuadro mostrado en la Figura 5.16.

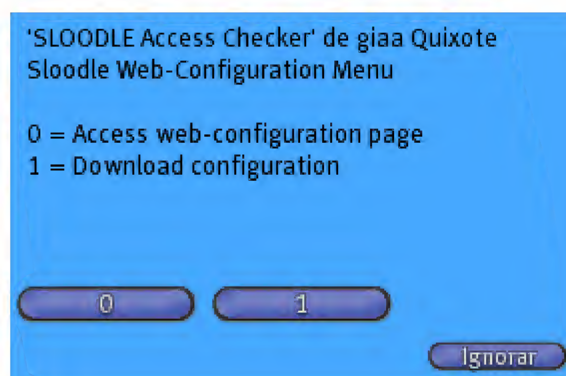


Figura 1306. Configuración de Access Checker (VI)

- Pulsando el botón *1* se nos indica que Access Checker se ha configurado correctamente (Figura 5.17).

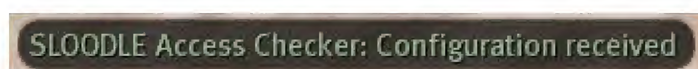


Figura 1315. Configuración de Access Checker (VII)

5.9.3. Pruebas de Access Checker

5.9.3.1. Utilización de un avatar registrado

Vamos a comprobar el Access Ckecker con un avatar registrado.

- Introducimos el avatar registrado dentro del "Access Ckecker", tal y como muestra la Figura 5.18:



Figura 1328. Utilización de Access Checker con un avatar registrado (I)

El Access Ckecker comprueba el registro del avatar en la plataforma Moodle.

- En el Chat nos muestra la siguiente información (Figura 5.19):

Figura 1339. Utilización de Access Checker con un avatar registrado (II)

Se nos indica que el acceso está garantizado para el avatar Giaa Quixote.

5.9.3.2. Utilización de un avatar no registrado

Comprobamos Access Ckecker con un avatar no registrado en nuestra plataforma Moodle.

- Nos disponemos a entrar con otro avatar que no está registrado en Moodle:



Figura 134. Utilización de Access Checker con un avatar no registrado (I)

- Sloodle nos muestra el siguiente mensaje en el Chat del avatar "Saga Bayn":

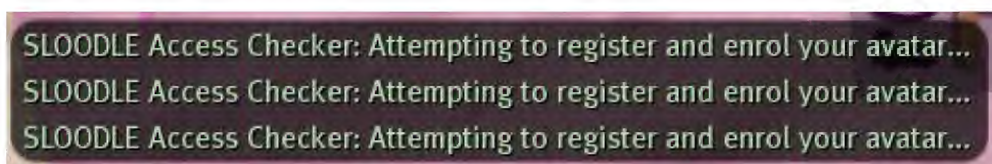


Figura 135. Utilización de Access Checker con un avatar no registrado (II)

- Vemos también el cuadro mostrado en la Figura 5.22:

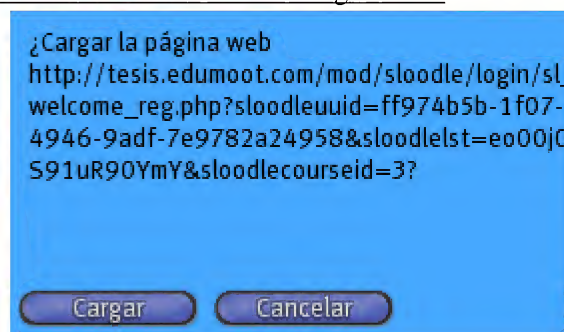


Figura 1362. Utilización de Access Checker con un avatar no registrado (III)

- Pulsando el botón "*Cargar*", accedemos a nuestro sitio Moodle (Figura 5.23).

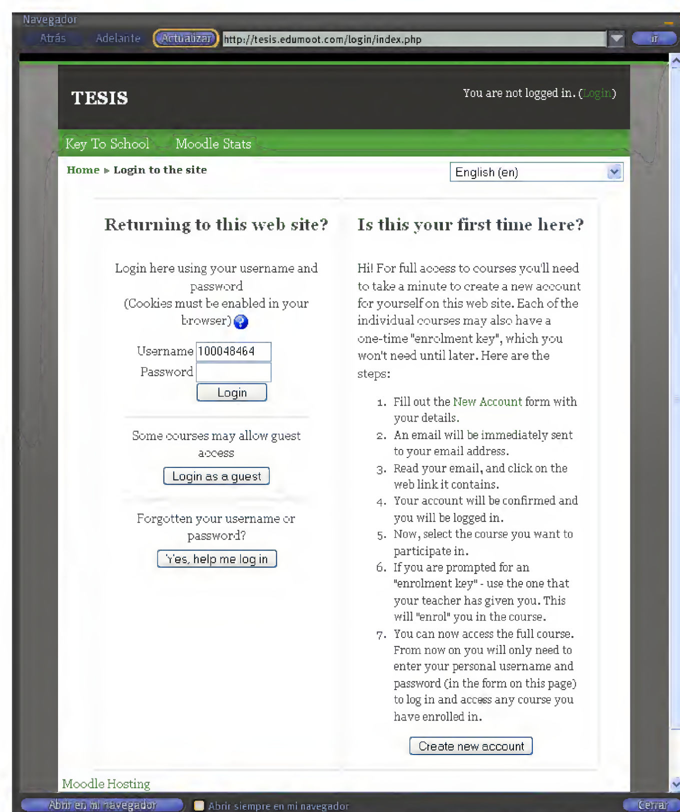


Figura 1373. Utilización de Access Checker con un avatar no registrado (IV)

- Seleccionamos la opción "*Create new Account* (crear una nueva cuenta)", con lo que accedemos a la página de registro (Figura 5.24). Insertamos nuestros datos para disponer así de un usuario registrado.

Home ▸ Login ▸ New account — English (en)

Choose your username and password

Username* saga

Password* keeD.qu4 ☑ Unmask

More details

Email address* e100048464@hotmail.com

Email (again)* e100048464@hotmail.com

First name* Eduardo

Surname* Rojo

City/town* Madrid

Country* Spain

Create my new account | Cancel

There are required fields in this form marked*.

Figura 1384. Utilización de Access Checker con un avatar no registrado (V)

Creamos una cuenta en nuestro sitio Moodle, para tener un usuario registrado, y hacer pruebas.

- Tras pulsar el botón *Create my new account*, visualizaremos la pantalla mostrada en la Figura 5.25.

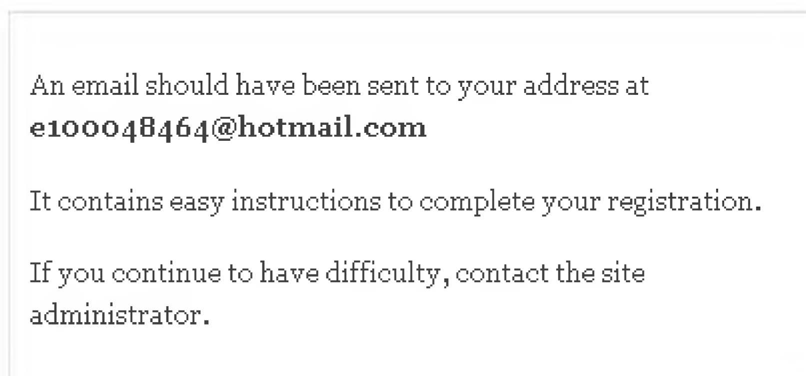


Figura 1395. Utilización de Access Checker con un avatar no registrado (VI)

- Pulsamos el botón *Continue* y revisamos a continuación el correo que habremos recibido:

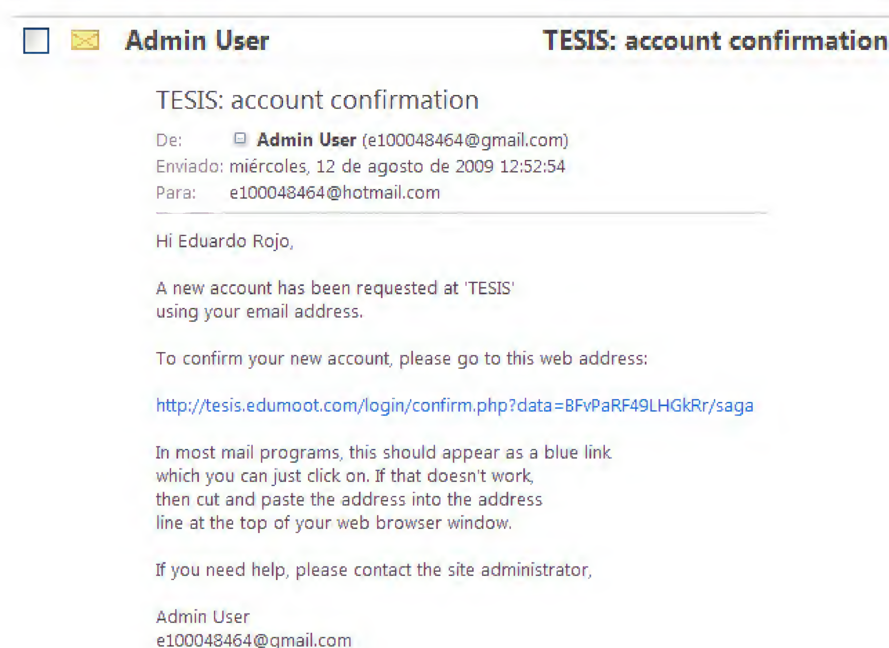


Figura 1406. Utilización de Access Checker con un avatar no registrado (V)

Nos informan que confirmemos nuestra cuenta y, de esta forma, asegurarse que el e-mail introducido al crear la cuenta sea correcto.

- Hacemos clic en la dirección web que nos indica para confirmar la cuenta. Con ello, se nos confirma que el registro se ha completado correctamente (Figura 5.25):

http://tesis.edumoot.com/login/confirm.php?data=BFvPaRF49LHGkRr/saga

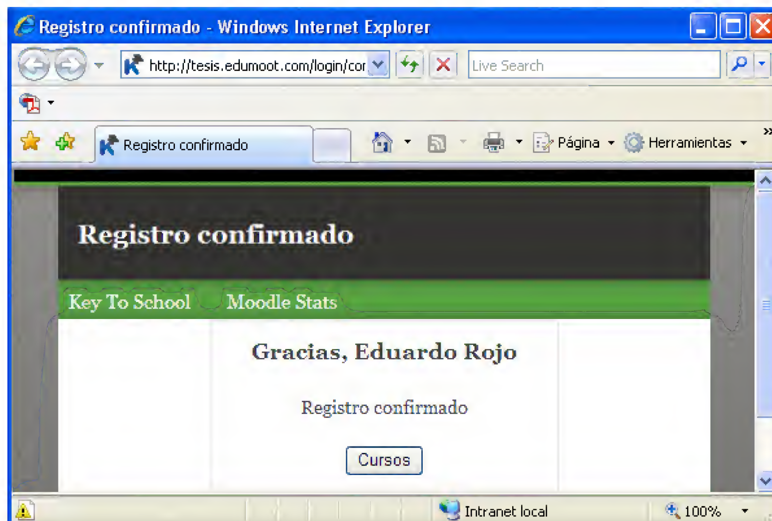


Figura 1415. Utilización de Access Checker con un avatar no registrado (VI)

5.10. Access Checker Door

5.10.1. Definición de Access Checker Door

Esta utilidad, realiza las mismas funciones que Sloodle Access Checker, pero en vez de activarse al entrar el avatar en una zona, se activa cuando el avatar toca el objeto correspondiente. Si al avatar dispone de permisos, éste les permitirá abrir la puerta de acceso.

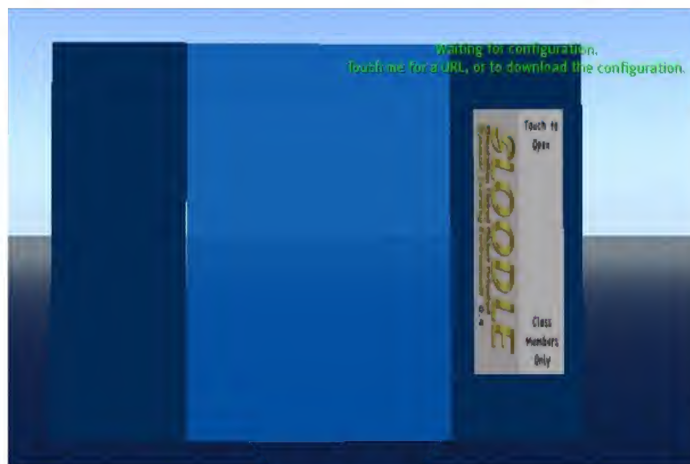


Figura 1428. Access Checker Door

5.10.2. Configuración de Access Checker Door

Vamos a configurar la herramienta Access Checker Door.

- <u>Hacemos clic sobre el **rez objects**</u>:



Figura 1439. Configuración de Access Checker Door (I)

- Nos muestra el siguiente menú:

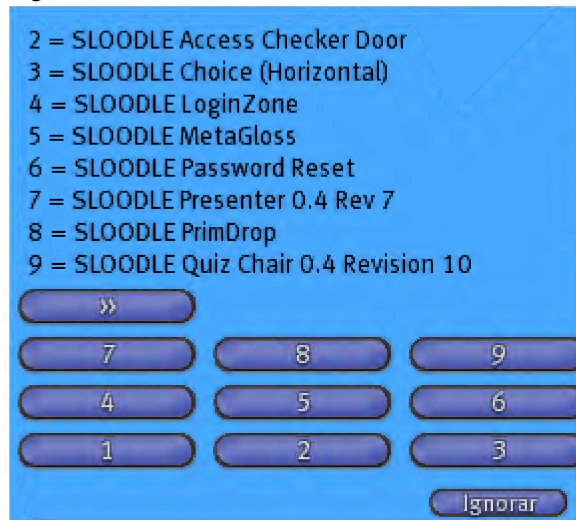


Figura 144. Configuración de Access Checker Door (II)

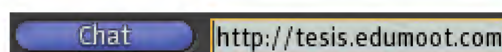
- Hacemos clic en el botón 2, con lo que visualizaremos Access Checker Door (Figura 5.28).

- Hacemos clic en *Access Ckecker Door*. Veremos un mensaje por el Chat indicando que introduzcamos los datos relativos a nuestro sitio Moodle.

SLOODLE Access Checker Door: Please chat the address of your Moodle site, without a trailing slash. For example: http://www.yoursite.blah/moodle

Figura 145. Configuración de Access Checker Door (III)

- Introducimos nuestros datos en el Chat y visualizaremos la ventana mostrada en la Figura 5.32.



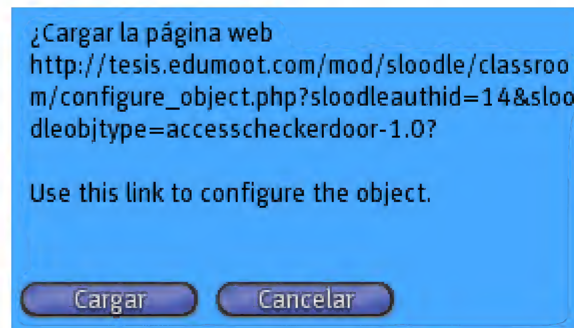


Figura 146. Configuración de Access Checker Door (IV)

- Pulsamos el botón *Cargar* y visualizaremos la ventana mostrada en la Figura 5.33.

Module ▸ SLOODLE Object Authorization

Object Details
Object Name: SLOODLE Access Checker Door
Object UUID: 1fdcbea5-5327-5bf5-c017-7fe3810cb1ea
Object Type: accesscheckerdoor-1.0

SLOODLE Object Authorization

Select Controller:

3. OpenSim y OsGrid
SLOODLE Controller: Controlador para OpenSim

Do you want to authorize this object? Yes

Submit

Figura 147. Configuración de Access Checker Door (V)

- Seleccionamos *SLOODLE Controller: Controlador para OpenSim y pulsamos el botón submit. Visualizaremos con ello la pantalla mostrada en la Figura 5.34*.

Home ▸ SLOODLE Module ▸ Object Configuration

Authorizing for: OpenSim y OsGrid > SLOODLE Controller: Controlador para OpenSim

Object authorization has been successful.

Object Details
Object Name: SLOODLE
Access Checker Door
Object UUID:
1fdcbea5-5327-5bf5-c017-7fe3810cb1ea
Object Type:
accesscheckerdoor-1.0

Object Configuration

Access Level

Object Access Level
This determines who may access the object in-world

Use object: Public

Submit

Figura 148. Configuración de Access Checker Door (VI)

- <u>Pulsando el botón *Submit*</u> se nos muestra la pantalla de la Figura 5.35.

Object Configuration

Object Details
Object Name: SLOODLE
Access Checker Door
Object UUID:
1fdcbea5-5327-5bf5-c017-7fe3810cb1ea
Object Type:
accesscheckerdoor-1.0
Authorized for: OpenSim y
OsGrid > SLOODLE
Controller: Controlador para
OpenSim

Number of settings stored: 2

Continue

Figura 149. Configuración de Access Checker Door (VII)

- Hacemos clic en el botón **Continue** y seguidamente en el **Access Checker Door**. Veremos la ventana mostrada en la Figura 5.36.

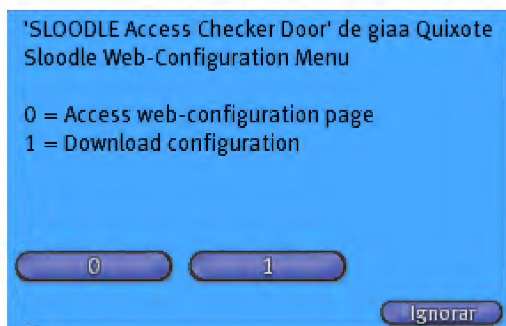


Figura 150. Configuración de Access Checker Door (VIII)

- Tras pulsar el botón 1 se nos informa que la configuración se ha realizado correctamente.

SLOODLE Access Checker Door: Configuration received

Figura 151. Configuración de Access Checker Door (IX)

5.10.3. Prueba de Access Checker Door

Vamos a probar Access Checker Door.

- Hacemos clic en el **Access Checker Door** (Figura 5.28) y se nos muestra el siguiente mensaje en el chat (Figura 5.38).

SLOODLE Access Checker Door: Attempting to register and enrol your avatar...
SLOODLE Access Checker Door: Thank you giaa Quixote. Your account was already enrolled.
SLOODLE Access Checker Door: Access is granted to giaa Quixote.

Figura 152. Prueba de Access Checker Door (I)

- Ahora, tras finalizar correctam,ente la configuración en el apartado anterior, podemos ver la puerta abierta para nuestro avatar



Figura 153. Prueba de Access Checker Door (II)

5.11. Choice Horizontal

5.11.1. Definición de Choice Horizontal

Esta herramienta permite al instructor adaptar sus actividades Choice MOODLE en Second Life, ofreciendo un medio para ejecutar las encuestas, las elecciones o la recopilación de información general de los participantes en un curso. Por lo tanto, una vez que una encuesta ha sido creada a través de la actividad Choice en Moodle, puede utilizarse esta utilidad para asignarla a un objeto Choice Sloodle en Second Life, disponiendo así de una representación grafica de la encuesta en el mundo 3D.



Figura 154. Choice Horizontal

5.11.2. Configuración de Choice Horizontal

Vamos a configurar el Choice Horizontal.

- Hacemos clic sobre el **rez objects**:



Figura 155. Configuración de Choice Horizontal (I)

- Nos muestra el siguiente menú:

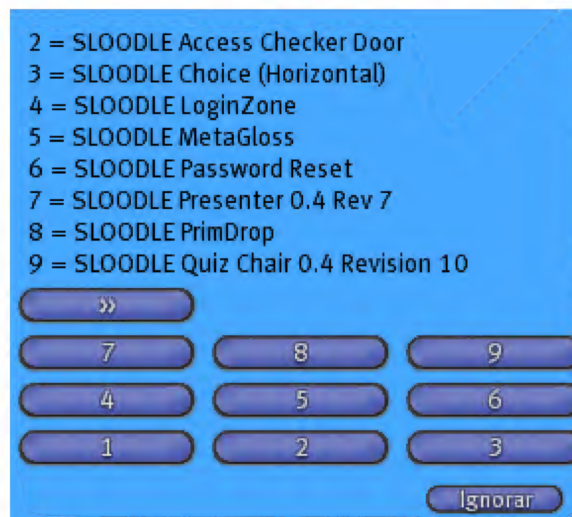


Figura 156. Configuración de Choice Horizontal (II)

- Hacemos clic en el botón **3**, visualizando **Choice (Horizontal)**:(Figura 5.41). Hacemos click en él, con lo que veremos la panatlla mostrada en la Figura 5.43.

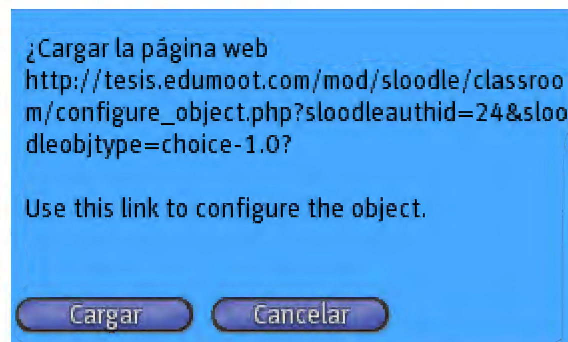


Figura 157. Configuración de Choice Horizontal (III)

- Pulsando **Cargar**, veremos la pantalla de la Figura 5.44.

TESIS

You are not logged in. (Login)

Key To School Moodle Stats

Home ▸ Login to the site

English (en)

Returning to this web site?

Login here using your username and password (Cookies must be enabled in your browser)

Username

Password

Login

Is this your first time here?

Hi! For full access to courses you'll need to take a minute to create a new account for yourself on this web site. Each of the individual courses may also have a one-time "enrolment key", which you won't need until later. Here are the steps:

Figura 158. Configuración de Choice Horizontal (IV)

- Introducimos nuestra información personal.

Returning to this web site?

Login here using your username and password (Cookies must be enabled in your browser)

Username 100048464

Password **********

Login

Figura 159. Configuración de Choice Horizontal (V)

- Nos muestra entonces la pantalla indicada en la Figura 5.46.

Object Configuration

General Configuration

Select Choice: Fecha de examen del Tema 1

Refresh time (seconds): 600

Show relative results?: No

SLOODLE Module ▸ Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE Choice (Horizontal)

Object UUID: 4c732aa2-44e4-ba33-e8f8-58363304d09b

Object Type: choice-1.0

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Server Access Level

This determines who may use the server resource

Access Level: Public

Submit

Figura 160. Configuración de Choice Horizontal (VI)

- Pulsando el botón **Submit, visualizaremos la pantalla mostrada en la Figura 5.47.**

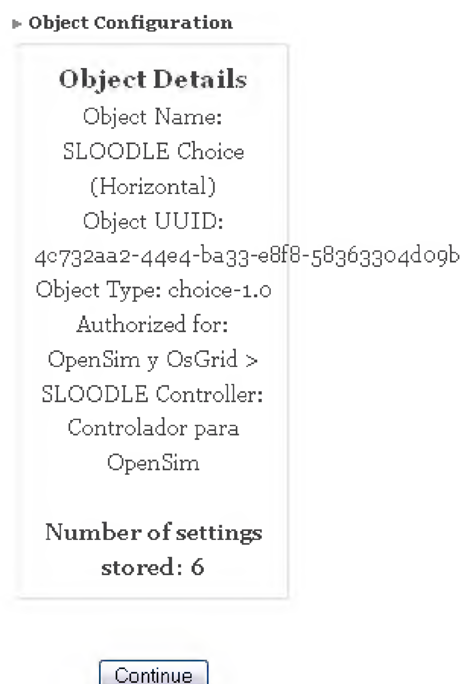


Figura 161. Configuración de Choice Horizontal (VI)

- Pulsamos el botón **Continue**, hacemos entonces clic en el **Choice Horizontal** y veremos el menú azul de la Figura 5.48.

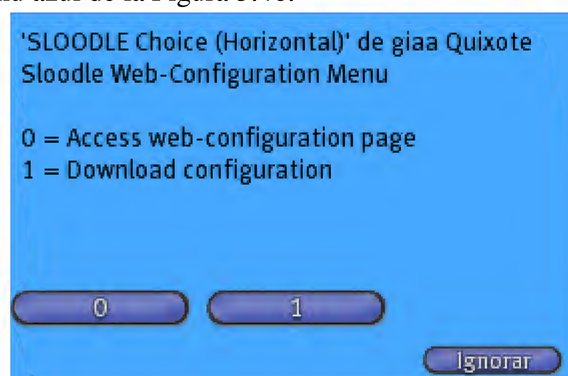


Figura 162. Configuración de Choice Horizontal (VII)

- Pulsamos el botón **1** y se nos muestra en el chat el siguiente mensaje.

SLOODLE Choice (Horizontal): Configuration received

Figura 163. Configuración de Choice Horizontal (VIII)

- Vemos el **Choice (Horizontal)**:



Figura 164. Configuración de Choice Horizontal (VIII)

Hacemos una prueba sobre una consulta a los alumnos sobre cuándo proponen que sea la fecha del primer examen de Open Sim, vemos que hay varia fechas, el 20 de enero, el 23 de enero, el 24 de enero y el 28 de enero, y vemos a la derecha las votaciones que hay sobre cada una de las pruebas.

5.12. Login Zone

5.12.1. Definición de Login Zone

Login zone es una herramienta alternativa para el registro de avatares. Crea una zona 3D en Second Life, de forma que cuando los usuarios hagan clic en el enlace correspondiente en Moodle, dispondrán de un enlace SLurl para teletransportarse a Second Life. Cuando su avatar aparezca en la zona, será registrado en su cuenta de usuario de Moodle.

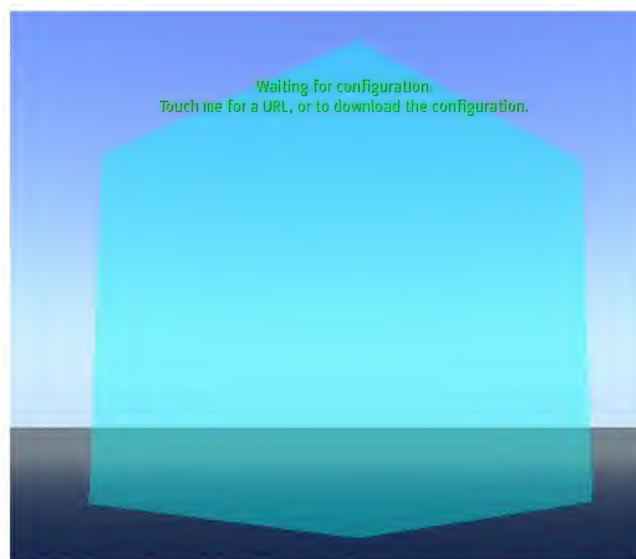


Figura 165. Login Zone

NOTA:
Es necesario tener activado el curso al cual quieres crear una zona de login:

Availability ? This course is available to students

También es necesario activar que los usuarios entren con clave, pues no tendría ningún sentido activar una clave para un curso, al cual se puede ya entrar sin clave.

Guest access ? Allow guests who have the key

5.12.2. Configuración de Login Zone

Vamos a configurar la herramienta Login Zone.

- <u>Hacemos clic sobre el **rez objects**</u>:



Figura 166. Configuración de Login Zone (I)

- <u>Nos muestra el siguiente menú</u>:

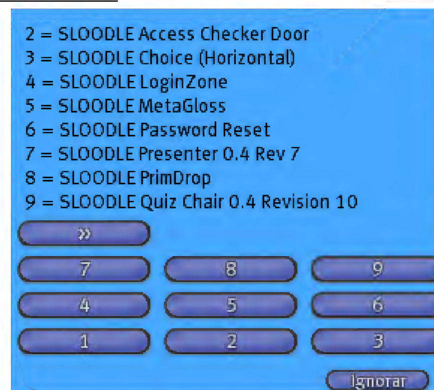


Figura 167. Configuración de Login Zone (II)

- Hacemos clic en el botón **4** y se nos muestra el **Login Zone**:



Figura 168. Configuración de Login Zone (III)

- Introducimos nuestros datos en el Chat:



- Hacemos clic en "Login Zone" y se nos muestra la siguiente pantalla:

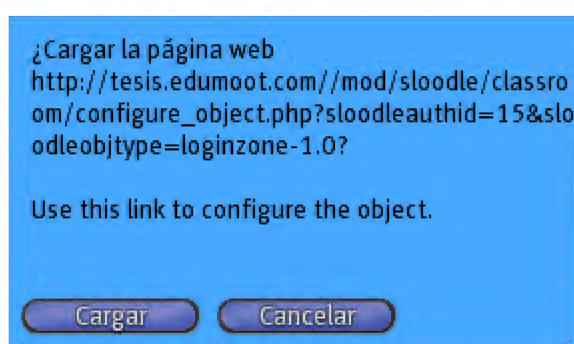


Figura 169. Configuración de Login Zone (IV)

- Hacemos clic en **Cargar** y se muestra la siguiente página:

SLOODLE Module ▸ SLOODLE Object Authorization

Object Details

Object Name: SLOODLE LoginZone

Object UUID: 2361d444-cb25-90fe-3123-325f5d5b5b8b

Object Type: loginzone-1.0

SLOODLE Object Authorization

Select Controller:

3. OpenSim y OsGrid

SLOODLE Controller: Controlador para OpenSim

Do you want to authorize this object? Yes

Submit

Figura 170. Configuración de Login Zone (V)

- Seleccionamos el **SLOODLE Controller: Controlador para OpenSim**:

3. OpenSim y OsGrid

SLOODLE Controller: Controlador para OpenSim

Figura 171. Configuración de Login Zone (VI)

- Hacemos clic en **Submit** y se muestra la siguiente pantalla:

Object authorization has been successful.

Object Details

Object Name: SLOODLE LoginZone

Object UUID: 2361d444-cb25-90fe-3123-325f5d5b5b8b

Object Type: loginzone-1.0

Object Configuration

General Configuration

Refresh time (seconds): 600

Submit

Figura 172. Configuración de Login Zone (VII)

Nos resta configurar los Objetos de Sloodle en nuestra plataforma Moodle.

- Hacemos clic en **Submit** y se nos muestra la siguiente pantalla:

Object Configuration

Object Details

Object Name: SLOODLE LoginZone

Object UUID: 2361d444-cb25-90fe-3123-325f5d5b5b8b

Object Type: loginzone-1.0

Authorized for: OpenSim y OsGrid > SLOODLE

Controller: Controlador para OpenSim

Number of settings stored: 2

Continue

Figura 173. Configuración de Login Zone (VIII)

Nos muestra los detalles de la configuración de los objetos Sloodle.

- Hacemos clic en el botón **Continue** y seguidamente en **Login Zone** (**Figura 5.54). Veremos la siguiente pantalla:**

'SLOODLE LoginZone' de giaa Quixote
Sloodle Web-Configuration Menu

0 = Access web-configuration page
1 = Download configuration

0 1

Ignorar

Figura 174. Configuración de Login Zone (IX)

- Hacemos clic en el botón **1**, El Chat nos muestra la siguiente información:

SLOODLE LoginZone: Configuration received

Figura 175. Configuración de Login Zone (X)

- Entramos en nuestro sitio Moodle como administrador. Seleccionamos un curso, por supuesto que tenga un controlador sloodle y que ese controlador esté asociado a la herramienta **Login Zone**:

OpenSim y OsGrid



- Y seleccionamos dentro del curso la opción **Turn editing on**:

Turn editing on

- Se nos muestra el siguiente **Blocks**:



- Y seleccionamos la opción “Sloodle Menu”:



- Y nos crea el siguiente menú en nuestro curso Moodle:

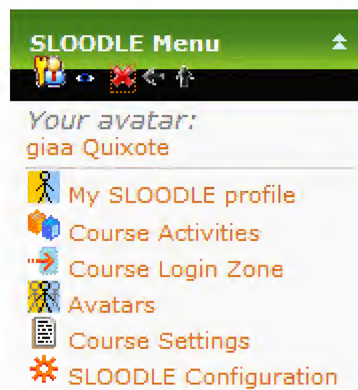


Figura 176. Configuración de Login Zone (XI)

5.12.3. Pruebas de Login Zone

5.12.3.1. Pruebas de Login Zone con avatar registrado

Vamos a entrar en Login Zone con un avatar registrado.

- Entramos dentro de **Login Zone**:



Figura 177. Prueba de Login Zone. Avatar no r5egistrado (I)

- El Chat nos muestra la siguiente información:

SLOODLE LoginZone: Thank you giaa Quixote. Your avatar was already authenticated.

Figura 178. Prueba de Login Zone. Avatar no registrado (II)

5.12.3.2. Prueba de Login Zone con un avatar no registrado

Vamos a hacer una prueba con Login Zone con un avatar no registrado.

- Entramos al sitio Moodle con la cuenta de un usuario registrado en el curso:

Nombre de usuario saga
Contraseña ••• Entrar

- Hemos entrado al curso de **OpenSim** y vemos el siguiente **SLOODLE Menu**:



- Hacemos clic en el enlace **Course Login zone** y veremos la siguiente ventana:

SLOODLE LoginZone

LoginZone Data

Position: <252,218,24>
Size: <10,10,10>
Region: Sloodle
Last Updated: 2010-01-16 03:21:37

Warning! This LoginZone data has not been updated recently, so it may no longer work.
The LoginZone may need to be rezzed in-world.

You do not have a LoginZone allocation yet. Please click the button below to get one.

Generate new LoginZone position

- Hacemos clic en "Generate new LoginZone position". Veremos la siguiente ventana:

SLOODLE LoginZone

LoginZone Data

Position: <252,218,24>
Size: <10,10,10>
Region: Sloodle
Last Updated: 2010-01-16 03:21:37

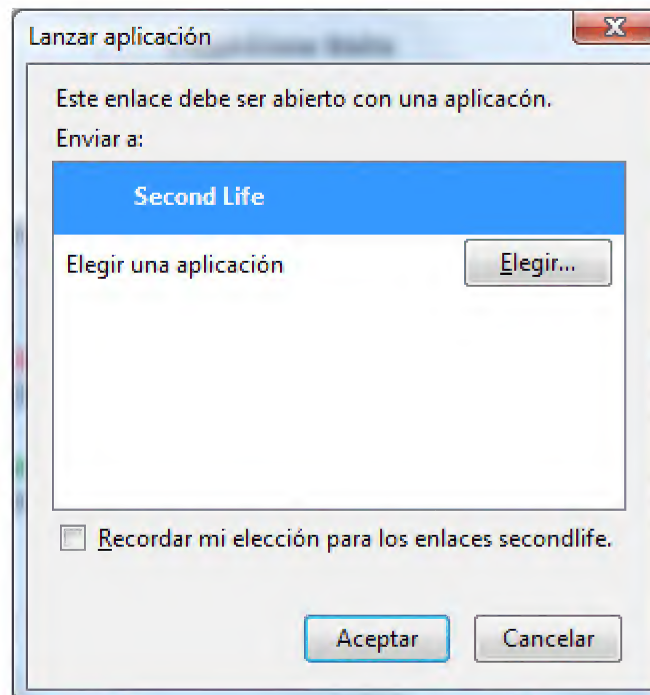
Warning! This LoginZone data has not been updated recently, so it may no longer work.
The LoginZone may need to be rezzed in-world.

Successfully allocated a LoginZone.
Please note that your Login Position will expire in 15 minutes. If you do not manage to use it in this time, then you will need to return here to re-activate it.

Click here to teleport to the Login Zone.

Generate new LoginZone position

- Hacemos clic en el enlace "Click here to teleport to the Login Zone". Se Nos abrirá la siguiente ventana:



* Pulsamos el botón **Aceptar**:

* Estamos en la siguiente posición:



Figura 179. Prueba de Login Zone. Avatar registrado (I)

Estamos en unas coordenadas de Second Life, en la isla de Sloodle.

* Vemos que se abre la siguiente ventana:

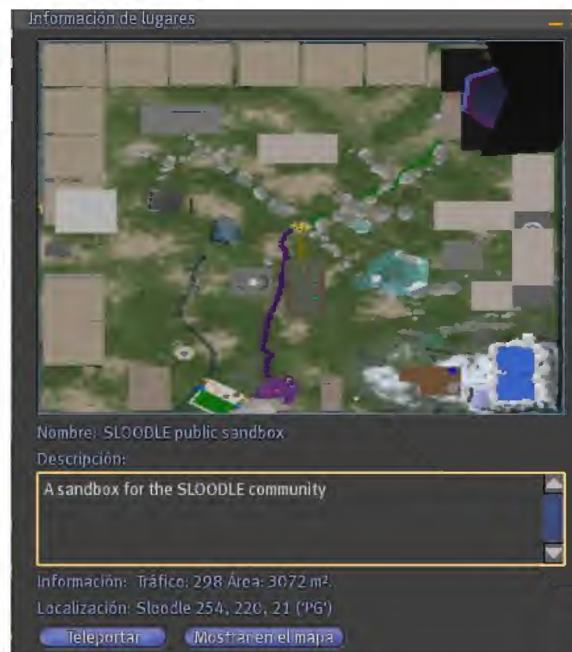


Figura 180. Prueba de Login Zone. Avatar registrado (II)

Panel que nos da opción para teletransportarnos.

* Hacemos clic en teletransportar:

- Nos muestra el menú que nos teletransporta:



Figura 181. Prueba de Login Zone. Avatar registrado (III)

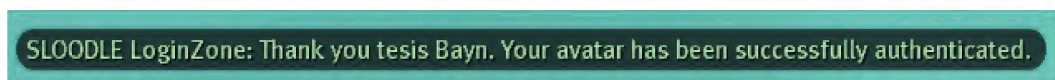
- Y nos encontramos en la posición de Second Life, donde esta nuestro sitio de enseñanza:



Figura 182. Prueba de Login Zone. Avatar registrado (IV)

El teletransporte nos ha llevado a la zona que le hemos indicado que sea la Login Zone.

- Y recibimos el siguiente mensaje:



5.13. MetaGloss

5.13.1. Definición de MetaGloss

Con esta utilidad, accedemos a los glosarios de Moodle desde el chat Second Life.



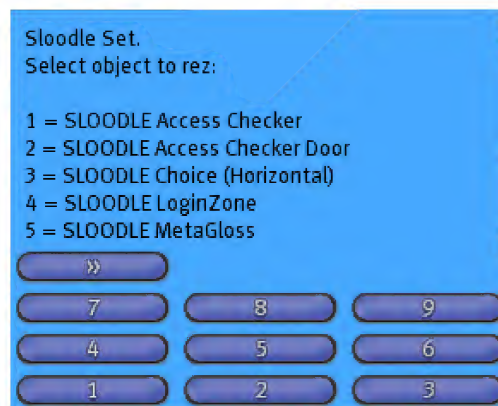
5.13.2. Configuración de MetaGloss

Vamos a configurar la herramienta MetaGloss.

- Hacemos clic sobre el **rez objects**:



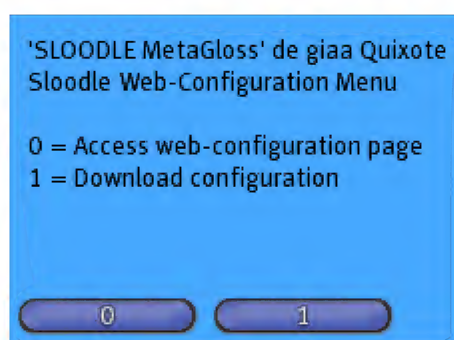
- Nos sale la siguiente ventana:



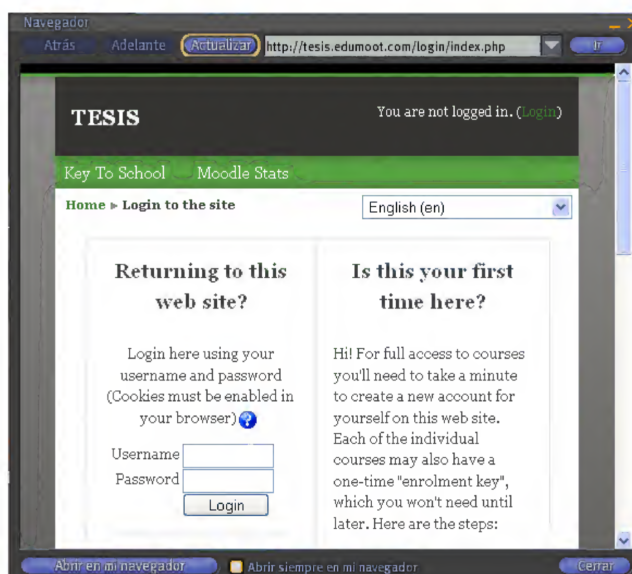
- Elegimos el botón **5**. Aparece un objeto **MetaGloss**:



- Hacemos clic sobre MetaGloss, y nos aparece la siguiente ventana:



- Hacemos clic en el botón **1.** Nos sale la siguiente ventana:



Introducimos nuestros datos de administrador.

- Nos sale la siguiente pantalla:

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Name: SLOODLE MetaGloss
Object UUID: e81cf876-d48a-6e0c-94d4-ce4212bbd73e
Object Type: glossary-1.0

Object Configuration

General Configuration

Select Glossary: Glosario de Accediendo a nuestra región.

Show Partial Matches: Sí

Search Aliases: No

Search Definitions: No

Idle timeout (seconds): 120

This determines who may access the object in-world

Use object: Public

Control object: Owner

Server Access Level

This determines who may use the server resource

Access Level: Public

- Ahora empezaremos la configuración del MetaGloss:

  - **Select Glossary** (Elección del Glosario):

    Selecciono el "Glosario de Descargas y carpetas", pues quiero acceder con este objeto al glosario del tema 1.

    Select Glossary: Glosario de Descargas y carpetas

  - **Show Partial Matches:**

    Que coincida en parte la palabra que hemos seleccionado para buscar en el glosario.

    Show Partial Matches: Sí

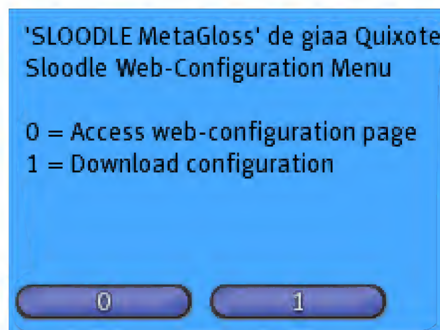
➢ **Idle timeout (seconds):**

Tiempo que se mantenga el MetaGloss si no lo usamos.

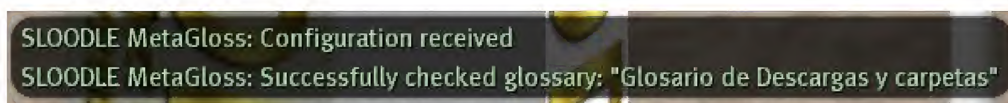
- Hacemos clic en Submit. Nos muestra la siguiente ventana:



- Hacemos clic en continuar y seguidamente en MetaGloss. Nos muestra la siguiente ventana azul:



- Hacemos clic en el botón **0** para bajarnos la configuración. Nos muestra el siguiente mensaje:



El mensaje nos indica que ha recibido la configuración, y que la configuración ha sido correcta para el glosario: “Glosario de Descargas y carpetas”.

5.13.3. Prueba de MetaGloss

- Introducimos en el chat:

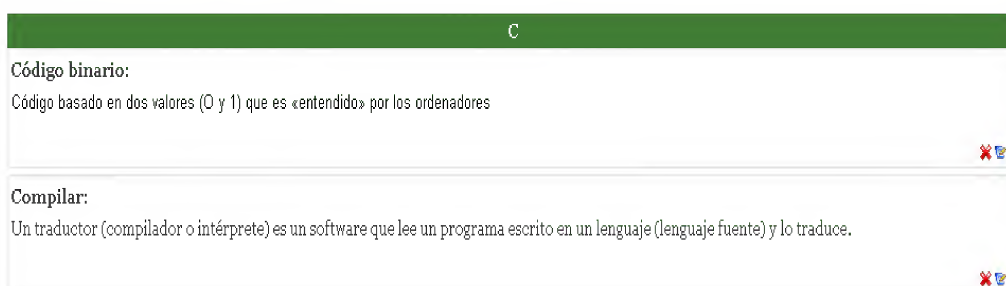
"/def" termino a buscar en el glosario

Vamos a comprobar el funcionamiento.

- Vamos a nuestro sitio Moodle, y hacemos clic en "Glosario de Descargas y carpetas":



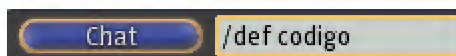
- Nos muestras los siguientes términos que están en nuestro introducidos en nuestro "glosario de Descargas y carpetas":


C

Código binario:
Código basado en dos valores (0 y 1) que es «entendido» por los ordenadores

Compilar:
Un traductor (compilador o intérprete) es un software que lee un programa escrito en un lenguaje (lenguaje fuente) y lo traduce.

- Vamos a "Second Life" y en el Chat escribimos "/def codigo":



- En Second Life nos da el siguiente resultado:

giaa Quixote: /def codigo
SLOODLE MetaGloss: Código binario = Código basado en dos valores (0 y 1) que es «entendido» por los ordenadores
SLOODLE MetaGloss: Number of definitions found for "codigo": 1

5.14. Password Reset

5.14.1. Definición de Password Reset

La herramienta de restablecimiento de contraseña, cuando se hace clic en ella, restablece la contraseña a los usuarios registrados, genera una contraseña temporal, y les envía un mensaje instantáneo con la nueva contraseña en la línea de chat que sólo se pueden ver en Second Life. Esto es útil sobre todo si tiene la inscripción automática (automatic enrollment) activada en Second Life.

Cuando la inscripción automática está activada, los estudiantes en Second Life no tienen que visitar su página web de Moodle a fin de empezar a usar sus Herramientas Sloodle. El único aspecto negativo de esto es que si ya tienen una cuenta de Moodle en su sitio Moodle, su avatar, no estarán conectadas a él, o si son nuevos usuarios, su dirección de correo electrónico no será capturado. Por lo tanto, sería preferible que los usuarios vincular sus cuentas de Avatar, sin la inscripción automática activada (auto-enrollment). Otro efecto secundario de esto, es si deciden

ingresar a su página web más adelante, se necesita primero recuperar su auto-generado contraseña. Pueden hacer esto haciendo clic en el [Sloodle Password Reset tool].



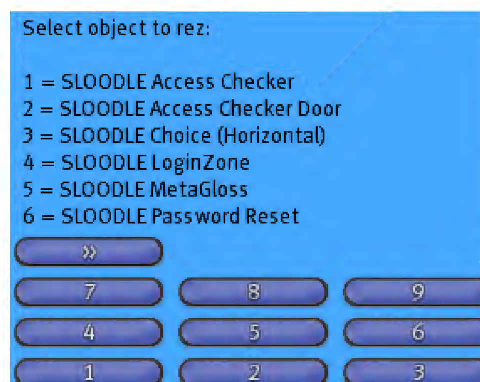
5.14.2. Configuración de Password Reset

Vamos a configurar Password Reset.

- Hacemos clic sobre el **rez objects**:



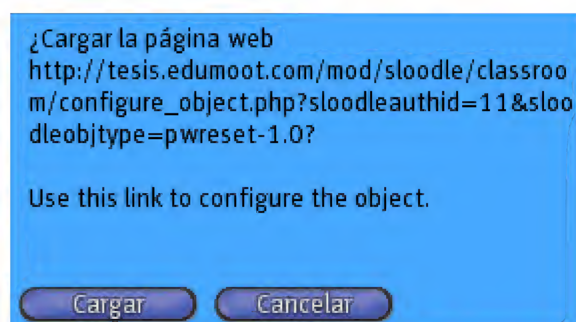
- Nos muestra el siguiente menú:



- Hacemos clic en el botón **6**. Nos muestra el **Moodle Password Reset**, y hacemos clic en él:



- Nos muestra el siguiente ventana azul:



- Hacemos clic en el botón **Cargar**. Nos muestra la siguiente página:

Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE Password Reset

Object UUID: 83da86ae-0d9d-30a2-3783-a1fbc895f483

Object Type: pwreset-1.0

Object Configuration

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Public
Group (SL)
Owner

Server Ac

This determines who may use the server resource

Access Level: Public

Submit

- Hacemos clic en el botón **Submit**. Nos muestra la siguiente ventana:

Object Configuration

Object Details

Object Name: SLOODLE Password Reset

Object UUID: 83da86ae-0d9d-30a2-3783-a1fbc895f483

Object Type: pwreset-1.0

Authorized for: OpenSim y OsGrid > SLOODLE Controller: Controlador para OpenSim

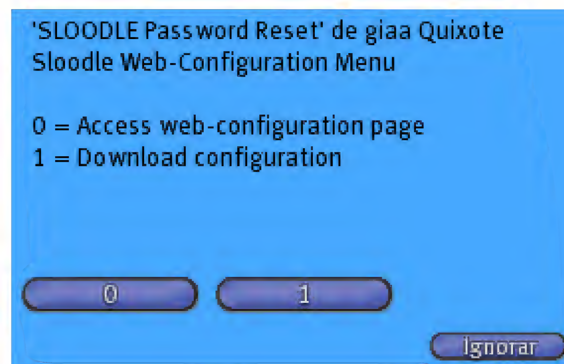
Number of settings stored: 3

Continuar

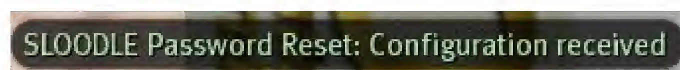
- Hacemos clic en el botón **Continuar** y seguidamente en **Moodle Password Reset**:



* Nos muestra la siguiente ventana azul:



* Hacemos clic en el botón **1**. Vemos que en el Chat nos sale el siguiente mensaje:



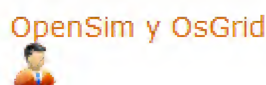
* Vemos el objeto **Moodle Password Reset**:



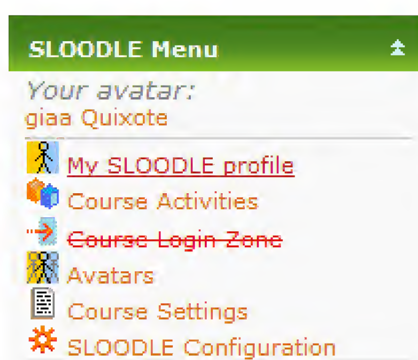
- Entramos como administrador a nuestro sitio Moodle:

Nombre de usuario 100048464
Contraseña ••••••••• Entrar

- Vamos al curso:



- Vamos al **Sloodle Menu**:



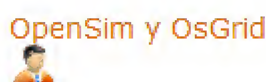
- Hacemos clic en el siguiente enlace, que es la configuración general de nuestro sitio Moodle:



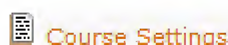
- Y comprobamos que “auto-enrolment” este puesto en “yes”, es decir permitimos que se nos auto-unan a cada uno de los cursos, por lo tanto registrara a un avatar aunque no esté registrado en nuestro sitio Moodle.

sloodle_allow_autoenrol Yes Default: No
Allow auto-enrolment for this site

- Hacemos clic en el curso de **OpenSim y OsGrid**:



- Hacemos clic en el siguiente enlace, en el cual configuramos un curso:



- Nos abre la siguiente página:

SLOODLE Course Configuration

(Course: "OpenSim y OsGrid")

On this page, you can configure the SLOODLE settings which affect your entire course. However, some of the settings may be disabled on your Moodle site by an administrator.

Please note: auto-registration and auto-enrolment are not suitable for all Moodle installations. Please read the documentation about each one before enabling them.

Course Settings

User Auto-Registration: Enabled

User Auto-Enrolment: Enabled

Submit

- Y miramos que "User Auto-Enrolment" este "Enabled":

User Auto-Enrolment: Enabled

5.14.3. Pruebas de Password Reset

5.14.3.1. Prueba de Password Reset con un avatar registrado

Existen dos métodos para autentificar al usuario con Sloodle.

1. **El método a través de la plataforma**.

Que es el recomendado. Cuando un objeto Sloodle encuentre un avatar en Second Life, que no han visto antes, le pedirá que se registre y le dará un enlace hacia nuestra plataforma.

2. **El método automático.**

El objeto Sloodle cuando encuentre a un avatar desconocido le creará una cuenta automáticamente, pero este avatar no estará registrado en la plataforma.



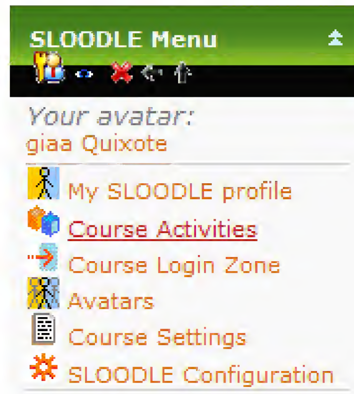
- Vamos a utilizamos el avatar "giaa Quixote" con el "PassWord Reset":

Nos saca un mensaje, indicándonos que no puede reajustar nuestra contraseña porque está asociada a una cuenta e-mail de Moodle. Y que si queremos reajustar nuestra contraseña, que lo hagamos desde nuestro sitio Moodle.

5.14.3.2. Pruebas de Password Reset con un avatar no registrado

Vamos a utilizamos el avatar “tesis Bayn” que no está registrado en nuestro sitio Moodle.

- Hacemos clic en **Sloodle Menu**:



- Hacemos clic en el enlace:



- Hacemos clic en **View all avatars**:

View all avatars

- Y vemos que efectivamente el avatar “tesis Bayn”, no pertenece a nuestro sitio Moodle:

All avatars

This lists all avatars for the entire site.

Avatar name	Avatar UUID	Last SLOODLE Activity	
giaa Quixote	23ea8218-73dd-41a6-823c-218fec422bf2	38 minutes ago	Delete
saga Bayn	ff974b5b-1f07-4946-9adf-7e9782a24958	20 hours ago	Delete

- Ahora nuestro avatar hace clic con el botón izquierdo del ratón en **Moodle Password Reset**:



- Sale el siguiente mensaje:

SLOODLE Password Reset: Sorry tesis Bayn. Error -321 occured while trying to reset your password.

Nos indica, que lo lamenta, pero que no puede resetear nuestra contraseña, pues nuestro avatar no está registrado en el sitio Moodle.

- Ahora, “tesis Bayn” utiliza la Quiz Chair:



- Y vemos que el avatar “tesis Bayn”, se ha registrado en nuestro curso, pero no tiene clave:

OpenSim y OsGrid

User	Avatar
tesis Bayn	tesis Bayn
eduardo rojo	saga Bayn

- El avatar “tesis Bayn” hace clic con el botón izquierdo del ratón en **Moodle Password Reset**:



- “Moodle Password Reset” nos devuelve una clave de usuario para que este avatar que estaba registrado en nuestro sitio Moodle, pueda entrar a nuestro sitio Moodle:

SLOODLE Password Reset: Thank you tesis Bayn. Your password has been successfully reset.
Site: http://tesis.edumoot.com
Username: tesisbayn
Password: zjFFruV3

5.15. Presenters

5.15.1. Definición de Presenters

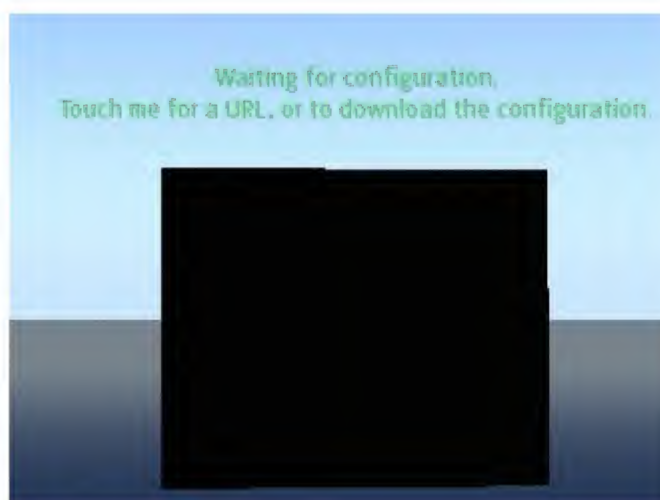
Presentaciones multimedia, en las que mezclan imágenes, videos y páginas Web, sin cargar las imágenes en Second Life.

El presentador Sloodle es una herramienta para crear presentaciones (para conferencias, seminarios o al propio ritmo uso tutorial) en Second Life. El presentador evita la obligación de cargar imágenes en Second Life y permite presentaciones que combinan imágenes, páginas web y vídeos.

Ya que el presenters utiliza la configuración multimedia de la parcela, el presenters debe ser propiedad del dueño de la tierra - o traspasada a los terratenientes del grupo donde la tierra es propiedad del grupo.

Las presentaciones se pueden ver en Second Life y / o en el propio Moodle.

El presenters puede estar configurado para que cualquier avatar - o sólo el propietario - pueda controlar la presentación con los botones adelante y atrás.

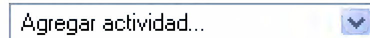


5.15.2. Configurar Presenters

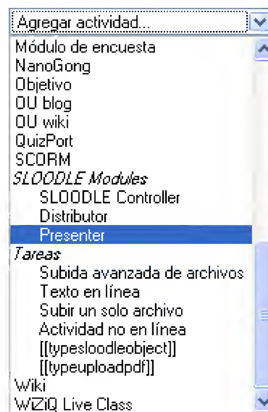
5.15.2.1. Añadir el modulo Presenter en Moodle

Vamos a añadir el módulo Presenter.

- Damos a agregar una actividad:



- Miramos los módulos y elegimos **Sloodle Modules / Presenter**:



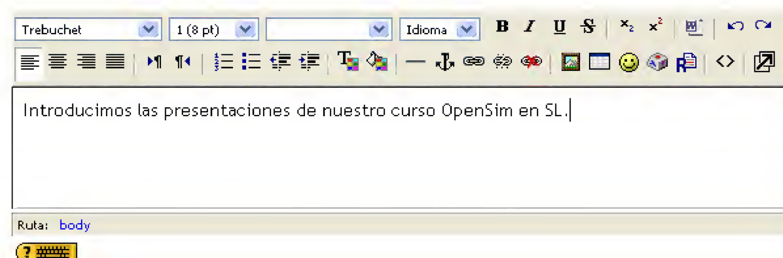
- Nos muestra una pantalla:

  - **Nombre**: Presentaciones OpenSim.

    Name* Presentaciones OpenSim

  - **Descripción:**

    Descripción



- Hacemos clic en Guardar cambios y mostrar:

  Guardar cambios y mostrar

5.15.2.2. Configurar el objeto Presenter

Vamos a configurar Presenter.

- Nos muestra la siguiente pantalla:

Presenter: Presentaciones OpenSim

Module Type: Presenter

Introducimos las presentaciones de nuestro curso OpenSim en SL.

No entries found.

Edit Presentation

View and delete entries

No entries found.

Add entry

URL:
Name:
Type: Web | Add entry

- Introduzco los datos siguientes:

  - **URL:**

    Introduzco una página web **http://opensimulator.org/wiki/Main_Page**

    URL: http://opensimulator.org/wiki/Main_Page

  - **NAME:** OpenSimulator.

    Name: OpenSimulator

  - **Type:** Introduzco "Web".

    Type: Web
    Image
    Video
    Web

- Damos al botón **Add entry**:

  Add entry

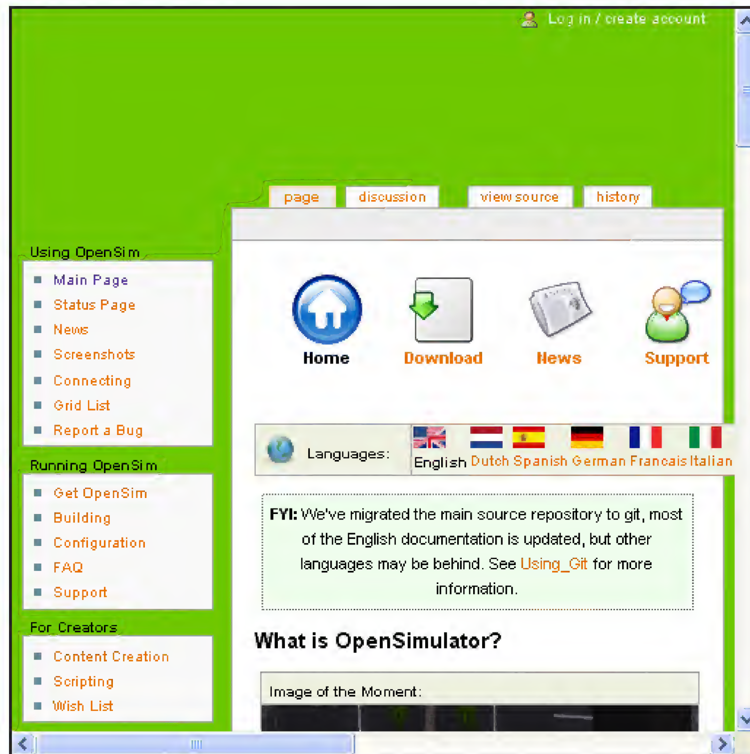
5.15.2.3. Viendo la presentación en Moodle

Vamos a ver la presentación en Moodle.

Presenter: Presentaciones OpenSim

Module Type: Presenter

Introducimos las presentaciones de nuestro curso OpenSim en SL.



If you cannot see the above entry, try this direct link instead.

"OpenSimulator"

← 1 of 1 →

Edit Presentation

View and delete entries

OpenSimulator (Web) ↑ ↓ Delete

Add entry

URL:
Name:
Type: Web Add entry

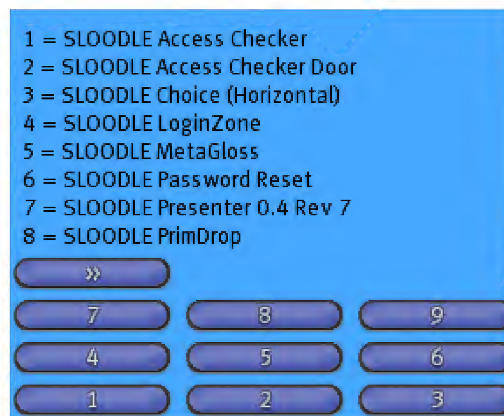
5.15.2.4. Configurar Presenter en Sloodle

Vamos a configurar Presenter en Sloodle.

- Hacemos clic sobre el **rez objects**:



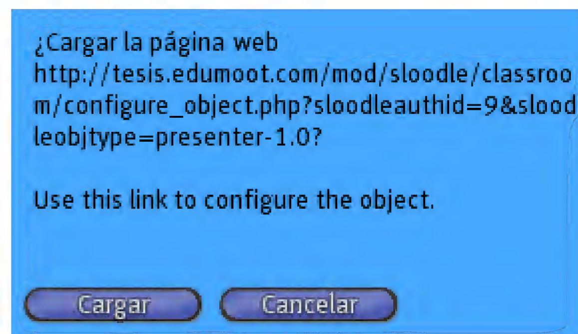
- Nos muestra la siguiente pantalla azul:



- Pulsamos el botón **7.** Nos aparece **Sloodle Presenter** y hacemos clic en Sloodle Presenter:



- Nos muestra el siguiente cuadro:



- Hacemos clic en el botón **Cargar**, nos muestra la siguiente página:

SLOODLE Module ▸ Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE Presenter 0.4 Rev 7

Object UUID: 8e0fcf23-589e-0c4d-1a6a-334db08f8987

Object Type: presenter-1.0

Object Configuration

General Configuration

Select Presenter:

Presenter: Presentaciones OpenSim

Access Level

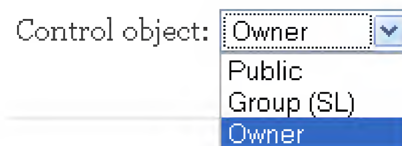
Object Access Level

This determines who may access the object in-world

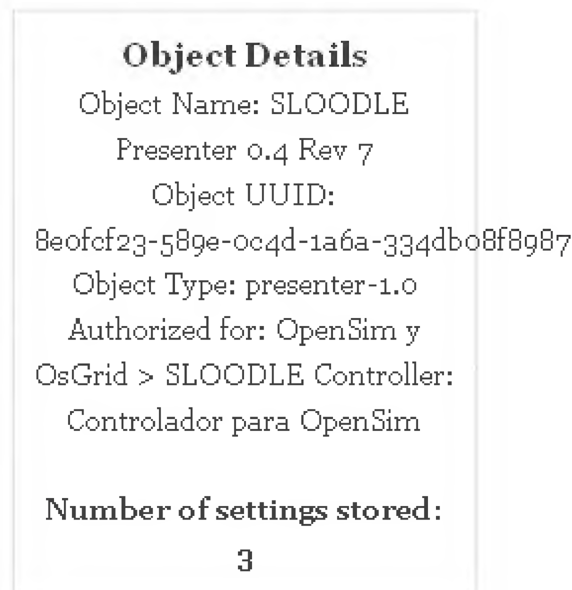
Control object: Owner

Submit

- Vemos las opciones de **Control object**:

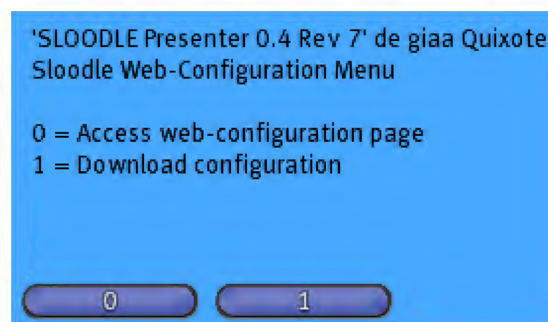


- Hacemos clic en el botón **Submit** y nos muestra la siguiente página:





- Hacemos clic en el botón **Continuar** y nos muestra el siguiente cuadro:



- Hacemos clic en el botón **1** y nos muestra la siguiente pantalla:

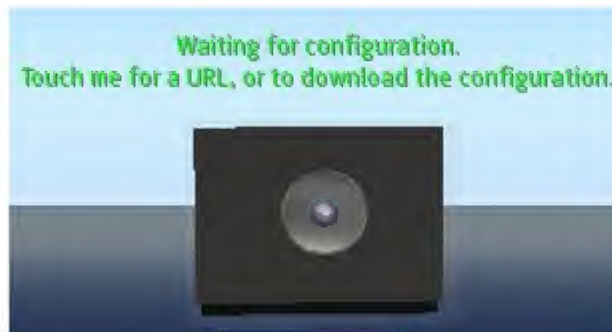


Nos indica que este presentador debe ser transferido a la parcela del propietario (del presenters) para mostrar su presentación.

5.16. PrimDrop

5.16.1. Definición de PrimDrop

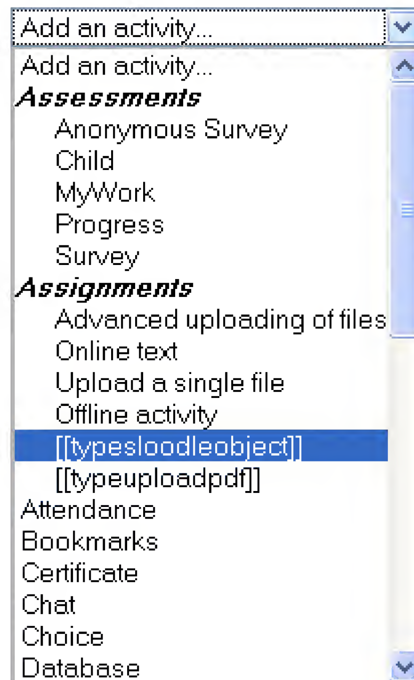
Esta utilidad, permite que sus estudiantes puedan enviar trabajos, entregar objetos de Second Life.



5.16.2. Configuración de PrimDrop

5.16.2.1. El profesor debe crear una tarea en su sitio Moodle

Vamos a ver como el profesor crea una tarea en su sitio Moodle.



- Elegimos la tarea **[[typessloodleobject]]**. Se nos abre una nueva ventana.

  - **Nombre:**

Assignment name* Entrega de objectos Second Life.

  - **Descripción:**

Description*

Trebuchet 1 (8 pt) Lang

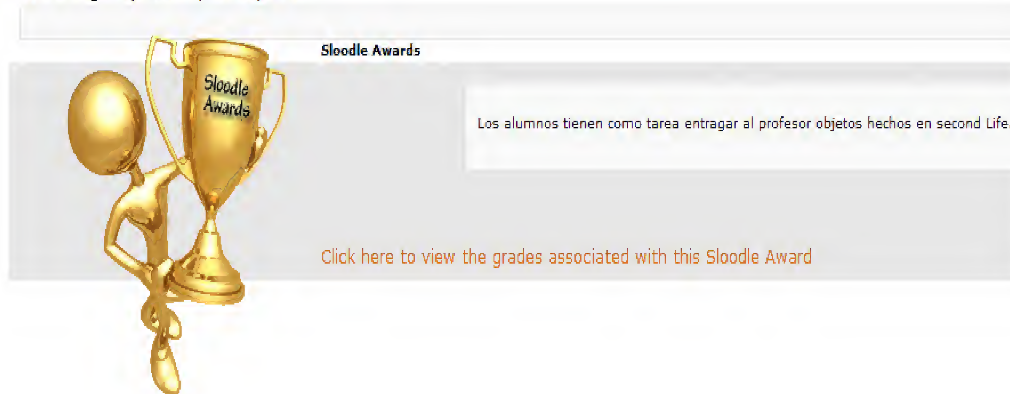
Los alumnos tienen como tarea entregar al profesor objectos hechos second life.

- Hacemos clic en el botón **Save and return to course**:

- Se nos ha creado el siguiente enlace:

Entrega de objetos Second Life.

- Y vemos la siguiente pantalla:



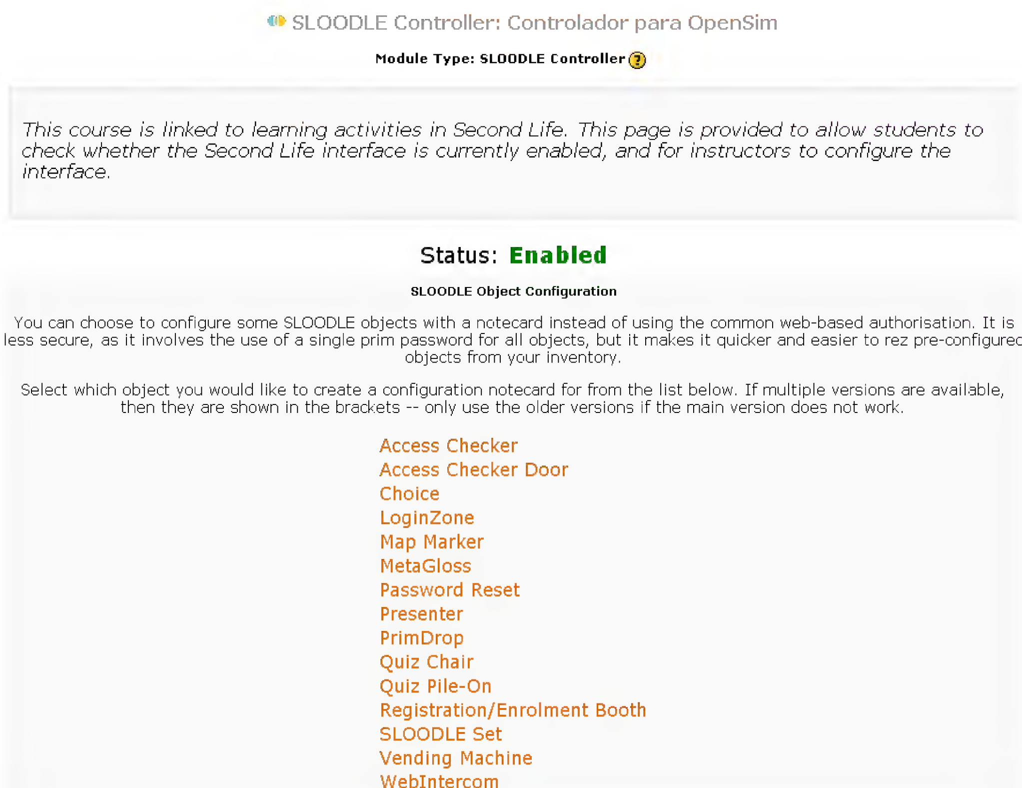
5.16.2.2. Haz clic en "Sloodle Controller", y genera una "NoteCard" para la actividad y el "PrimDrop" creados

Vamos a generar una NoteCard.

- Hacemos clic en **SLOODLE Controller: Controlador para OpenSim**:



- Nos aparece la siguiente pantalla:



Podemos ver los distintos objetos que se pueden programar en Sloodle.

- Hacemos clic en **Sloodle Set**:

SLOODLE Set

- Nos aparece la siguiente ventana:

SLOODLE Object Configuration

SLOODLE Set 1.0

No additional configuration options for this object.

Generate Notecard

- Hacemos clic en el botón **Generate Notecard**. Nos muestra la siguiente pantalla:

SLOODLE Configuration Notecard

SLOODLE Set 1.0

To configure a SLOODLE object, edit or create a notecard called 'sloodle_config' in its inventory, and add the text from the box below.

For security reasons, you should make sure that the 'sloodle_config' notecard *and* the object itself cannot be modified by the next owner.

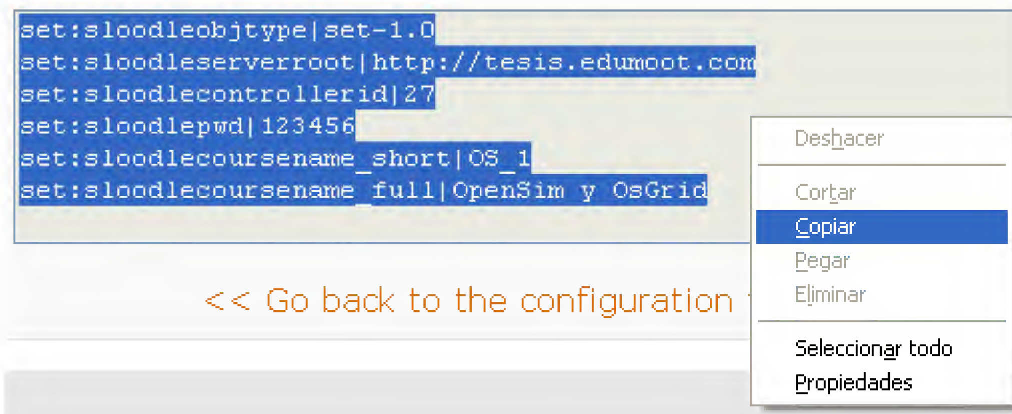
```
set:sloodleobjtype|set-1.0
set:sloodleserverroot|http://tesis.edumoot.com
set:sloodlecontrollerid|27
set:sloodlepwd|123456
set:sloodlecoursename_short|OS_1
set:sloodlecoursename_full|OpenSim y OsGrid
```

<< Go back to the configuration form.

5.16.2.3. Copie el texto generado dentro de un "Notecard" en Second Life llamado sloodle_config

Vamos a copiar el texto generado dentro de un Notecard.

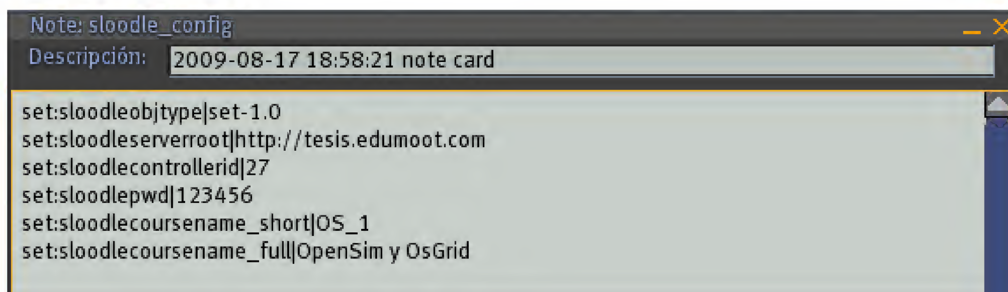
- Copio el texto de la NoteCard de Sloodle:



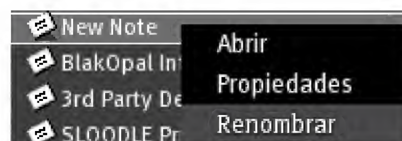
- Vamos a Second Life, hacemos clic en **Inventario**. Hacemos clic en "Crear / Nota nueva":



- Se nos abre una nota nueva, donde copiamos el contenido de "Generate Notecard":



- Hacemos clic en **Guardar**. Seleccionamos el "NoteCard", y damos a Renombrar:



- Y lo renombramos como "sloodle_config":



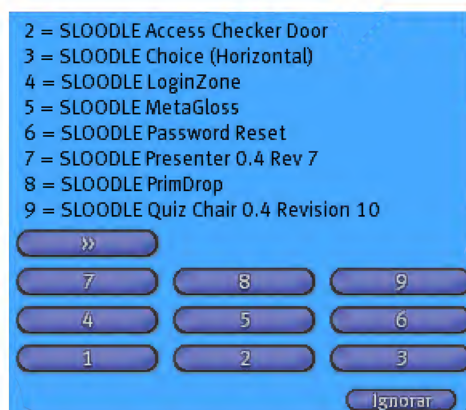
5.16.2.4. Arrastro un nuevo objeto “PrimDrop” dentro de Second Life desde de mi Second Life inventario

Vamos a crear el PrimDrop.

- Hacemos clic sobre el **rez objects**:



- <u>Nos muestra el siguiente menú azul</u>:



- <u>Pulsamos el botón **8**. Hemos creado el **PrimDrop**:</u>



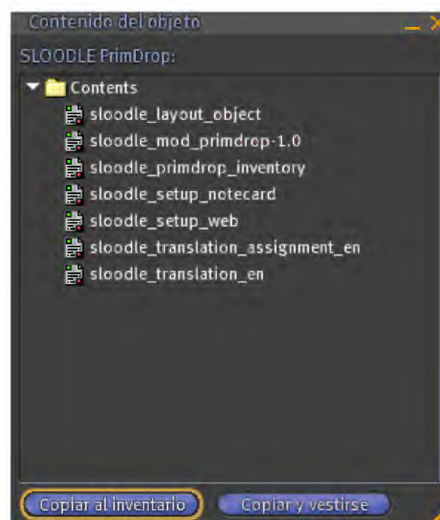
5.16.2.5. Arrastro la configuración de la "Notecard" de sloodle_config dentro del objeto "PrimDrop" en Second Life

Vamos a configurar el PrimDrop.

- Hacemos clic con el botón derecho del ratón, y hacemos clic en **Abrir**:



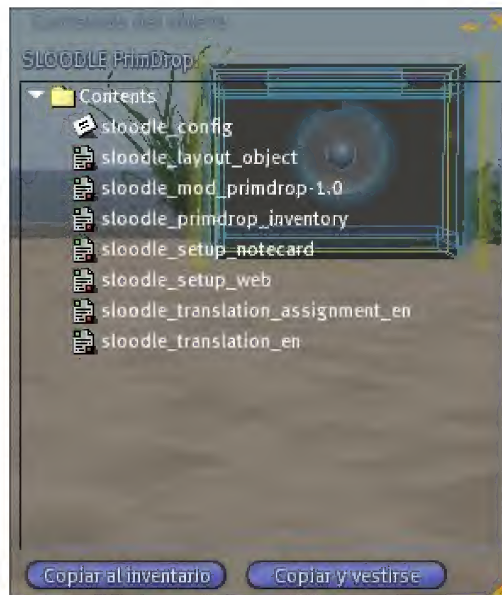
- Nos muestra la siguiente pantalla:



- Hago clic en mi **inventario**. Selecciono la Notecard **Sloodle_config**:



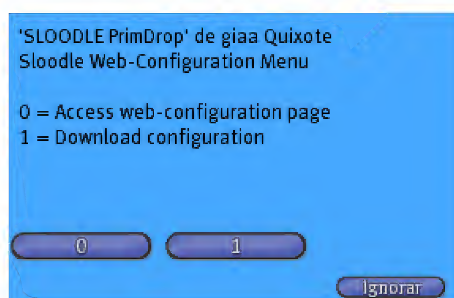
- La arrastro al "contenido" del objeto "PrimDrop":



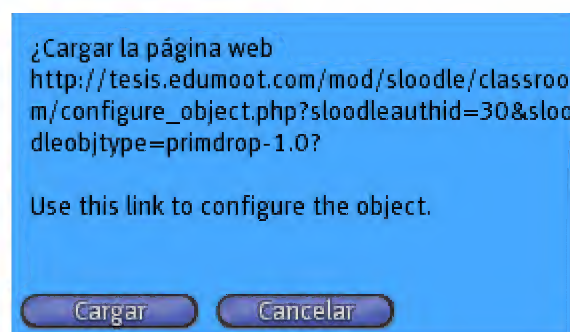
- Hago clic sobre el **PrimDrop**:



- Nos aparece la siguiente ventana azul:



- Hago clic en el botón **0**. Nos aparece la siguiente ventana azul:



- Hago clic en cargar. Nos abre la siguiente ventana:

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details
Object Name: SLOODLE PrimDrop
Object UUID: d5b0c53f-24da-9e35-8ed3-57a7f4ce0126
Object Type: primdrop-1.0

Object Configuration

General Configuration

Select Assignment: Entrega de objectos Second Life.

Access Level

Object Access Level
This determines who may access the object in-world

Use object: Public

Control object: Owner

Server Access Level
This determines who may use the server resource

Access Level: Public

Submit

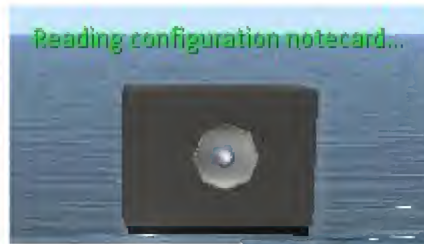
- Hacemos clic en **Submit.** Nos abre la siguiente pantalla:

Object Details
Object Name: SLOODLE PrimDrop
Object UUID: d5b0c53f-24da-9e35-8ed3-57a7f4ce0126
Object Type: primdrop-1.0
Authorized for: OpenSim y OsGrid > SLOODLE Controller: Controlador para OpenSim

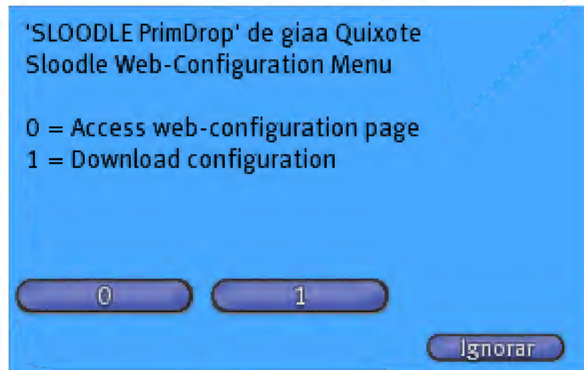
Number of settings stored: 5

Continue

- Hacemos clic en el botón **Continue** y en el **PrimDrop**:



- Se abre el siguiente menú azul:



- Hacemos clic en el botón **1**. En el Chat se muestra el siguiente mensaje:



- El "PrimDrop" ya está configurado:



5.16.3. Pruebas de PrimDrop

5.16.3.1. Pruebas de PrimDrop estudiante

Vamos a **enviar una tarea**.

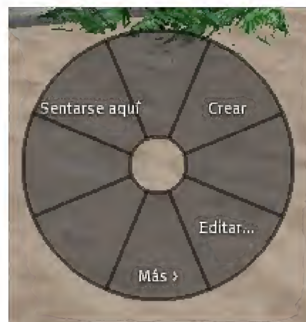
Entramos como un estudiante que está registrado tanto en él como su avatar en el sitio Moodle en ese curso.

- Entramos con el usuario "Eduardo rojo", que tiene el avatar "saga Bayn":

User	Avatar
eduardo rojo	saga Bayn

1. **Enviar una tarea, un estudiante debe crear un objeto en Second Life.**

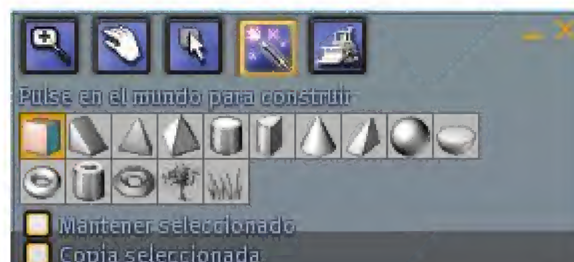
➢ Seleccionamos el botón derecho del ratón, y nos aparece el siguiente menú:



➢ Y hacemos clic en el botón **Crear**:



➢ Elegimos uno de los objetos para crear:



➢ Elegimos el cubo:



➢ Elegimos una zona para crear el cubo:

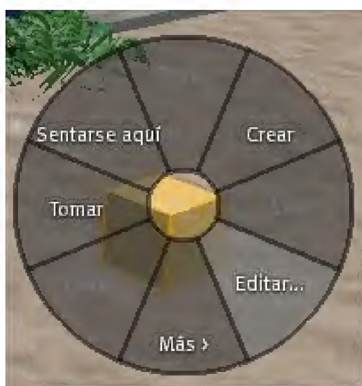


2. El estudiante debe fijar los permisos del objeto como “copiar/transferir”.

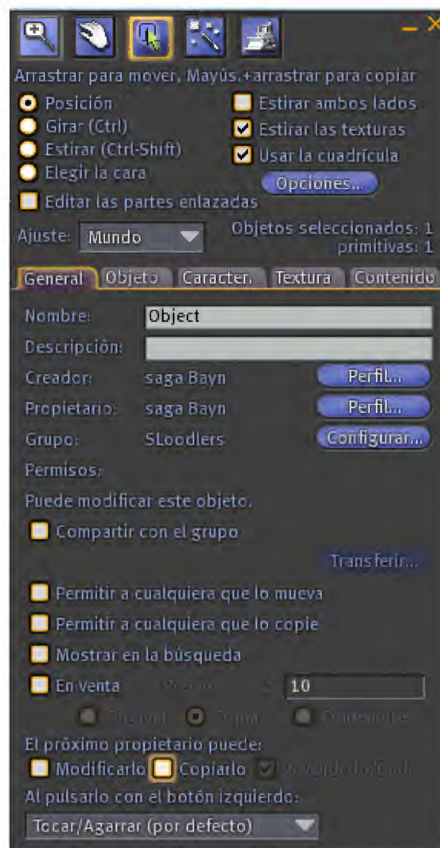


El estudiante puede hacer esto bien usando la ventana de edición cuando el objeto se cree, o después de que el objeto este dentro del inventario, haciendo clic en el objeto en el inventario y selecciona “propiedades”.

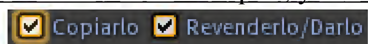
➢ Seleccionamos editar:



➢ Se nos abre la siguiente ventana:



➢ <u>Y seleccionamos Copiar, y automáticamente se encenderá "Revenderlo/Darlo"</u>:

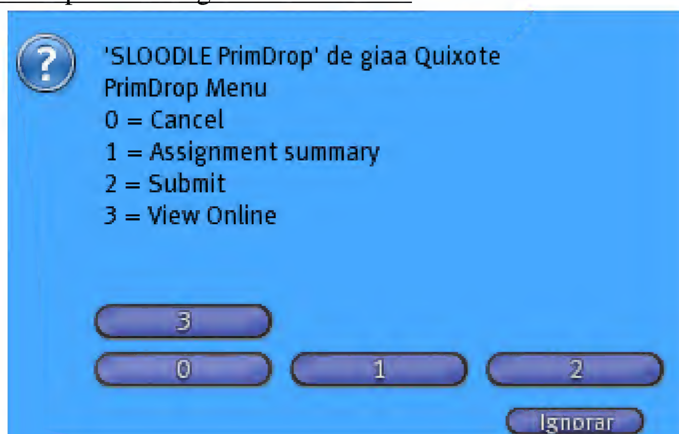


3. **Los estudiantes entonces harán clic en el objeto PrimDrop y seleccionarán "Enviar / Submit" desde el menú azul que aparece.**

➢ <u>Hacemos clic en el **PrimDrop**</u>:



➢ Nos aparece el siguiente menú azul:



➢ Hacemos clic en el botón 2 ("Submit"), Nos sale este mensaje:



4. **El estudiante seleccionara el objeto y presionara control + arrastrara el objeto hacia el PrimDrop (El PrimDrop se tornará rojo).**

Si salió bien, se enviara un mensaje:



5.16.3.2. Pruebas de PrimDrop profesor

Ahora recogemos como profesor o administrador los objetos que los alumnos nos han enviado.

- Entramos a Second Life como "giaa Quixote" que es el administrador:



- Hacemos clic con el botón derecho del ratón en el PrimDrop:



- Seleccionamos del menú, la opción **Editar**:



- Nos sale el siguiente menú:



- Y vemos el objeto que nos acaba de enviar el alumno “saga Bayn”:

Object (no modify)

5.17. Quiz Chair

5.17.1. Definición de Quiz Chair

Con esta utilidad, el avatar puede realizar los cuestionarios del curso elegido desde SL. Tiene aún algunas limitaciones, como el número de caracteres que podrán poner el avatar en las respuestas y el número de caracteres en las preguntas.



5.17.2. Configuración de Quiz Chair

Vamos a configurar el Quiz Chair.

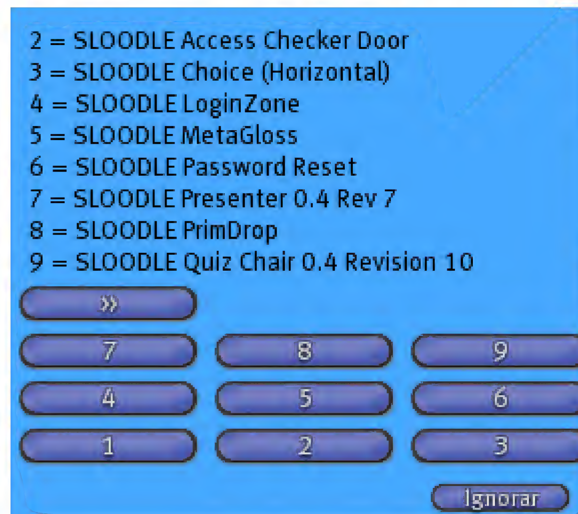
- <u>Los **requisitos previos**</u>:

  Tenemos que tener instalado en Moodle un Quiz.

- Hacemos clic sobre el **rez objects**:



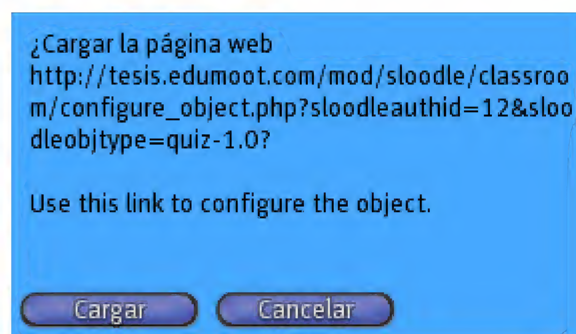
- <u>Nos muestra el siguiente menú</u>:



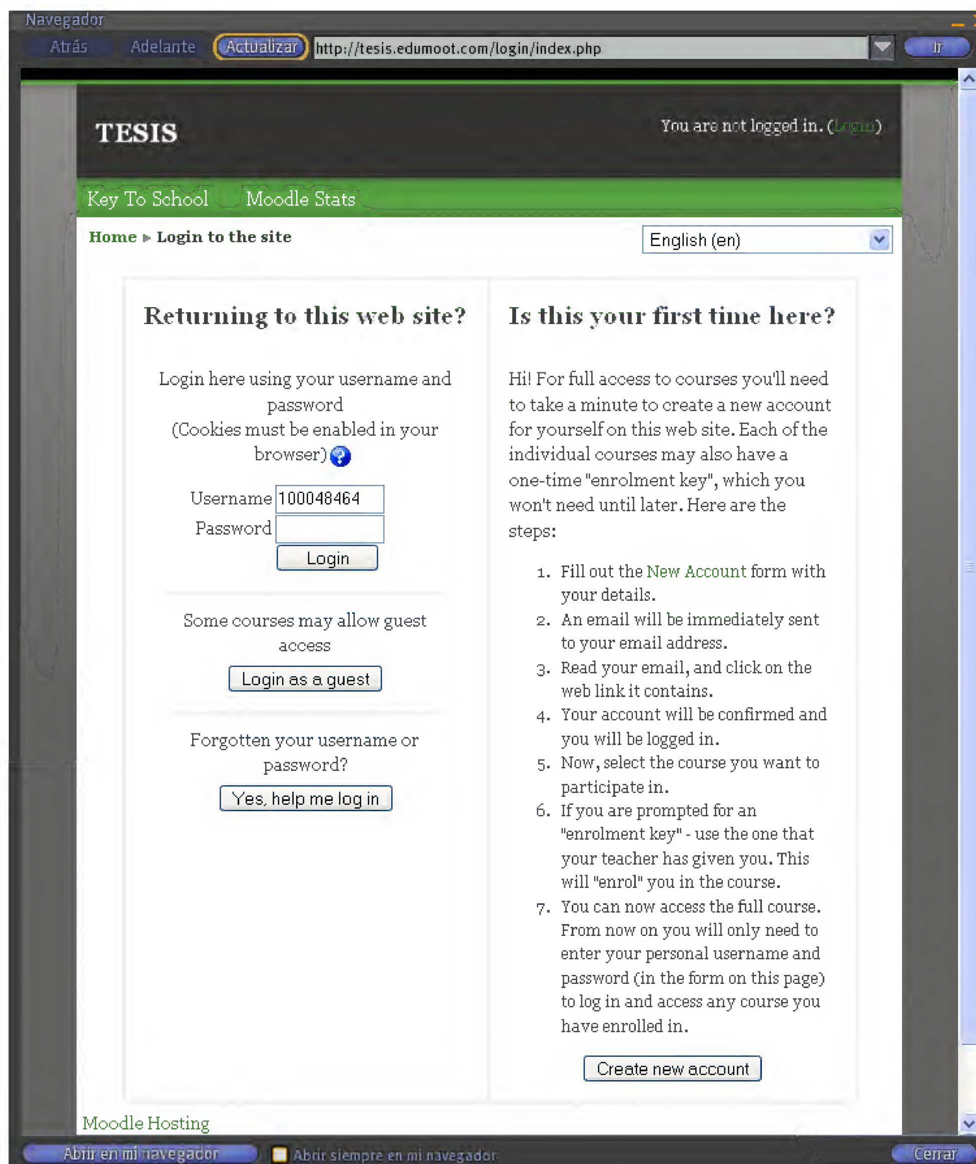
- Hacemos clic en el botón **9**, nos muestra el **Quiz Chair**:



- Hacemos clic en **Quiz Chair**, nos muestra el siguiente cuadro:



- Hacemos clic en el botón **Cargar**. Nos muestra la siguiente página:



- Introducimos nuestros datos y hacemos clic en el botón **Login**. Nos muestra la siguiente página:

SLOODLE Module ▸ Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE Quiz Chair 0.4 Revision 10

Object UUID: 7288c0b0-5a75-7e6a-3bd2-7e100fb082f0

Object Type: quiz-1.0

Object Configuration

General Configuration

Select Quiz: Quiz Tema 1. Descargas y carpetas.

Automatically repeat the quiz? No

Randomize question order? Yes

Use dialogs (instead of chat)? Yes

Play sounds? No

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Server Access Level

This determines who may use the server resource

Access Level: Public

Submit

- Hacemos clic en el botón **Submit**. Nos muestra la siguiente página:

Object Configuration

Object Details

Object Name: SLOODLE Quiz Chair 0.4 Revision 10

Object UUID: 7288c0b0-5a75-7e6a-3bd2-7e100fb082f0

Object Type: quiz-1.0

Authorized for: OpenSim y OsGrid > SLOODLE

Controller: Controlador para OpenSim

Number of settings stored: 8

Continue

- Hacemos clic en el botón **Continue y** en **Quiz Chair**:



- Nos muestra la siguiente pantalla:

'SLOODLE Quiz Chair 0.4 Revision 10' de giaa Quixote
Sloodle Web-Configuration Menu

0 = Access web-configuration page
1 = Download configuration

0 1

Ignorar

- Hacemos clic en el botón **1**. Vemos que en el Chat nos sale el siguiente mensaje:

SLOODLE Quiz Chair 0.4 Revision 10: Configuration received

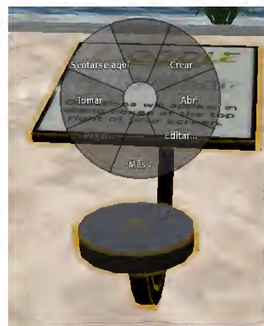
- Vemos el objeto **Quiz Chair**:



5.17.3. Pruebas de Quiz Chair

Vamos a hacer pruebas de la herramienta Quiz Chair.

- Pulsamos el botón derecho sobre la "Quiz Chair", y elegimos la opción "Sentarse aquí":



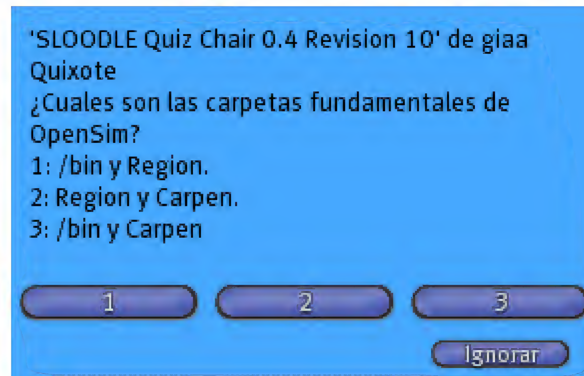
- El Chat nos muestra los siguientes mensajes:

SLOODLE Quiz Chair 0.4 Revision 10: Starting quiz for giaa Quixote
SLOODLE Quiz Chair 0.4 Revision 10: Fetching quiz data...
SLOODLE Quiz Chair 0.4 Revision 10: Ready to attempt: Quiz Tema 1. Descargas y carpetas..

- Ya estamos sentados en la **Quiz Chair**:



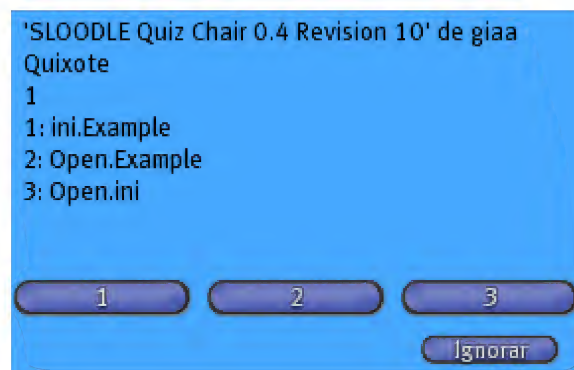
- Nos muestra el siguiente mensaje:



- Hacemos clic en el botón **1**: (**opción correcta**) El Chat nos muestra el siguiente mensaje:

SLOODLE Quiz Chair 0.4 Revision 10 susurra: Correct giaa Quixote.

- Nos muestra el siguiente mensaje:



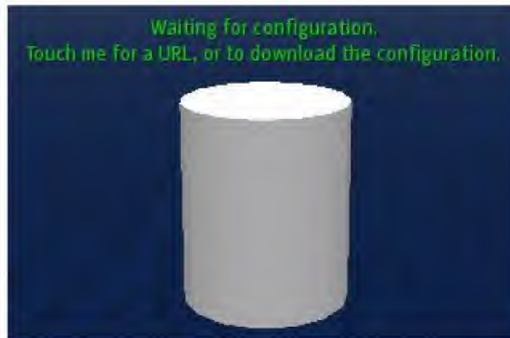
- Hacemos clic en el botón **2**: (**opción incorrecta**) El Chat nos muestra el siguiente mensaje:

SLOODLE Quiz Chair 0.4 Revision 10: Quiz complete giaa Quixote. Your final score was 1/2.

5.18. Quiz Pile On

5.18.1. Definición de Quiz Pile On

Permite a los usuarios en Second Life hacer un cuestionario de opción múltiple que el maestro ha creado en Moodle. Las preguntas se muestran como texto que flota por encima de la pila, y las posibles respuestas se muestran como texto que flota por encima de las semi-esferas que están rezzed (creadas) para cada pregunta. Los estudiantes se sientan en la Esfera que piensan que es la respuesta correcta. Entonces el cilindro (Quiz Pile-on) es tocado, y se hace clic en la opción “answer”. Si el usuario está sentado en la semi-esfera que tiene la respuesta equivocada, el usuario se caerá. Esta es una manera muy divertida para hacer un cuestionario.



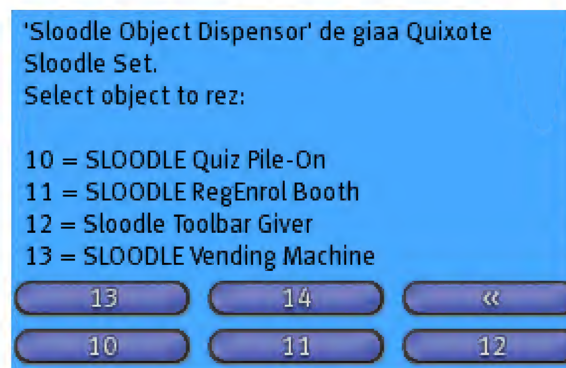
5.18.2. Configuración de Quiz Pile

Vamos a configurar Quiz Pile.

- Hacemos clic sobre el **rez objects**:



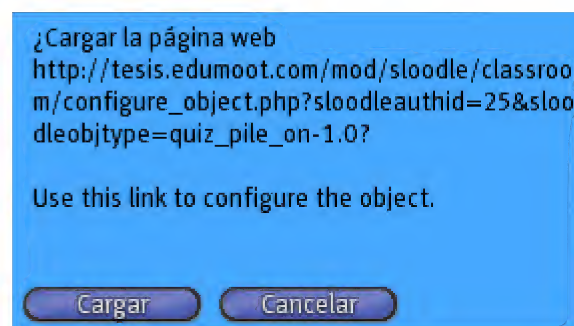
- Nos muestra el siguiente menú:



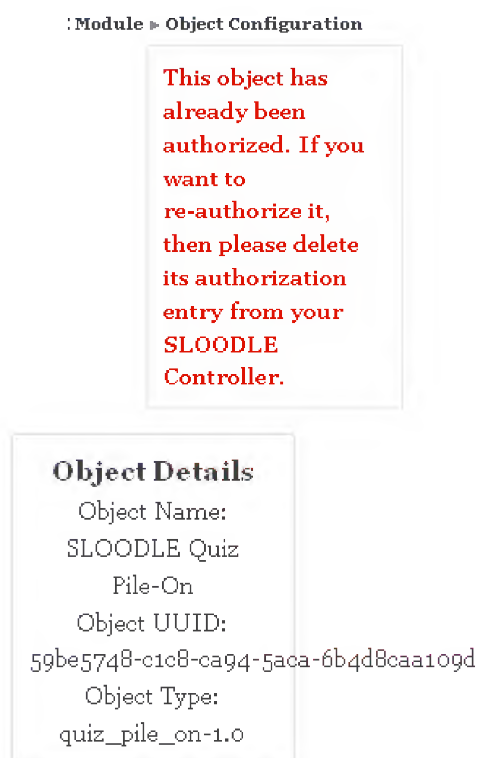
- Hacemos clic en el botón **10.** Nos muestra el **Quiz Pile-On** y hacemos clic en él:



- <u>Nos muestra el siguiente menú azul</u>:



- Hacemos clic en el botón **Cargar.** Nos muestra la siguiente pantalla:



Object Configuration

General Configuration

Select Quiz:

Quiz Tema 1. Descargas y carpetas.

Automatically repeat the quiz?

No

Randomize question order?

Yes

Play sounds? No

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Control object: Owner

Server Access Level

This determines who may use the server resource

Access Level: Public

Submit

- Hacemos clic en el botón **Submit**. Nos muestra la siguiente pantalla:

▸ Object Configuration

Object Details
Object Name:
SLOODLE Quiz
Pile-On
Object UUID:
59be5748-c1c8-ca94-5aca-6b4d8caa109d
Object Type:
quiz_pile_on-1.0
Authorized for:
OpenSim y OsGrid >
SLOODLE Controller:
Controlador para
OpenSim

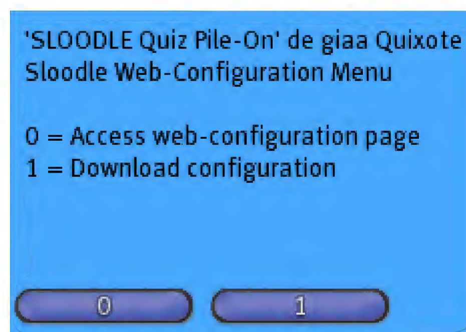
Number of settings stored: 8

Continue

- Hacemos clic en el **Quiz Pile-On**:



- Nos muestra el siguiente menú azul:



- Hacemos clic en el botón **1.** En el chat nos sale el siguiente mensaje:

SLOODLE Quiz Pile-On: Configuration received

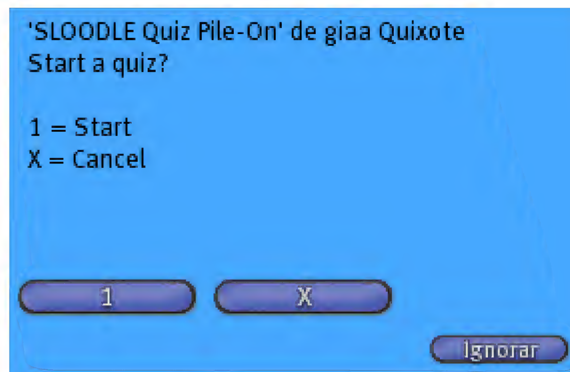
5.18.3. Prueba de Quiz Pile On

Vamos a probar el Quiz Pile On.

- Hacemos clic en el **Quiz Pile-On**:



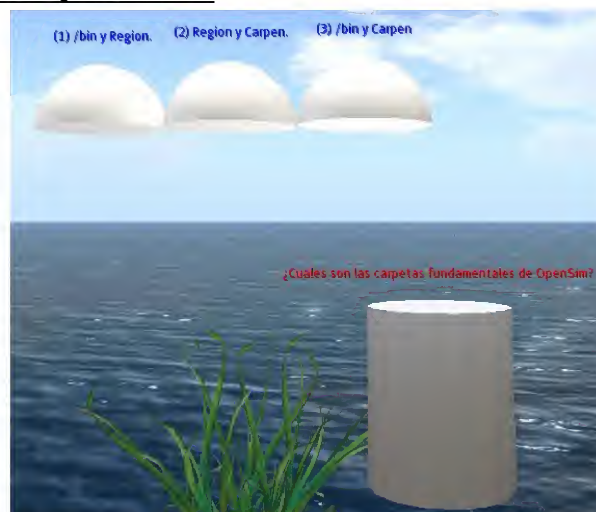
- Nos sale el siguiente menú azul:



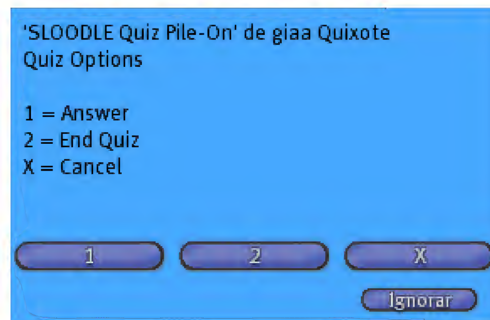
- Hacemos clic en el botón **1**. El chat nos muestra el siguiente mensaje:

SLOODLE Quiz Pile-On: Starting quiz for giaa Quixote
SLOODLE Quiz Pile-On: Fetching quiz data...
SLOODLE Quiz Pile-On: Ready to attempt: Quiz Tema 1. Descargas y carpetas..
SLOODLE Quiz Pile-On: ¿Cuales son las carpetas fundamentales de OpenSim?

- Ahora miramos el **Quiz Pile-On**:



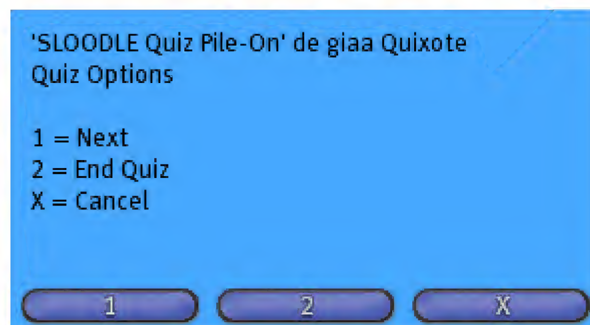
- Hacemos clic en el **Quiz Pile-On**, y nos muestra el siguiente menú azul:



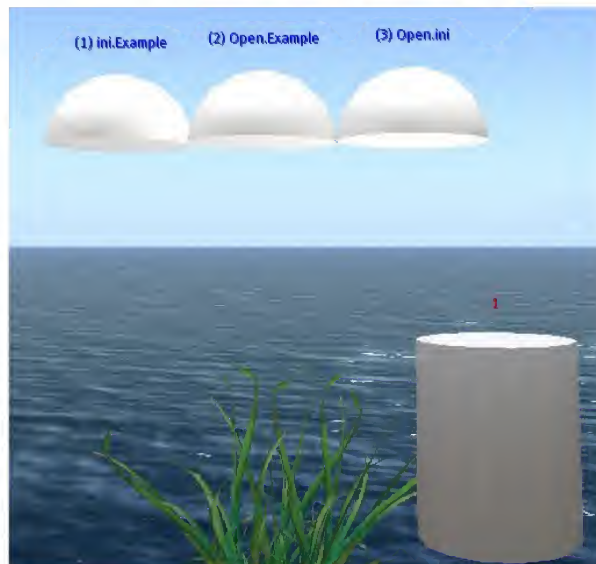
- Hacemos clic en el botón **1**.Nos muestra lo siguiente y hacemos clic en el **Quiz Pile-On**:



- Nos muestra el siguiente cuadro azul:

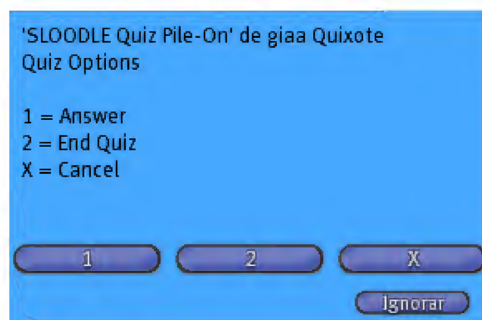


- Hacemos clic en el botón "1". Ahora miramos el "Quiz Pile-On":

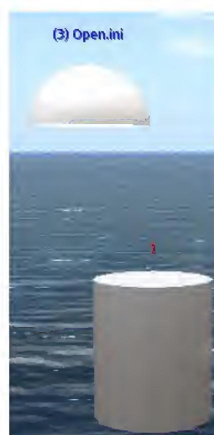


El juego de Quiz Pile on consiste en este caso, en que los alumnos se sienten en la semi-esfera que consideren que posee la respuesta correcta, y si fallan la respuesta, la semi-esfera desaparece con lo que el alumno se cae.

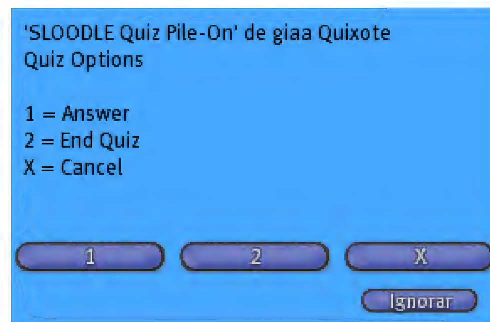
- Hacemos clic en el "Quiz Pile-On". Nos muestra el siguiente cuadro azul:



- Hacemos clic en el botón "1". Ahora miramos el "Quiz Pile-On":



- Hacemos clic en el "Quiz Pile-On". Nos muestra el siguiente cuadro azul:



- Hacemos clic en el botón "1". El chat nos muestra el siguiente mensaje:

SLOODLE Quiz Pile-On: Quiz complete.

5.19. RegEnrol_Booth

5.19.1. Definición de RegEnrol_Booth

Esta utilidad, permite a un usuario registrado de un sitio Moodle, registrar a su avatar en un curso de Moodle.



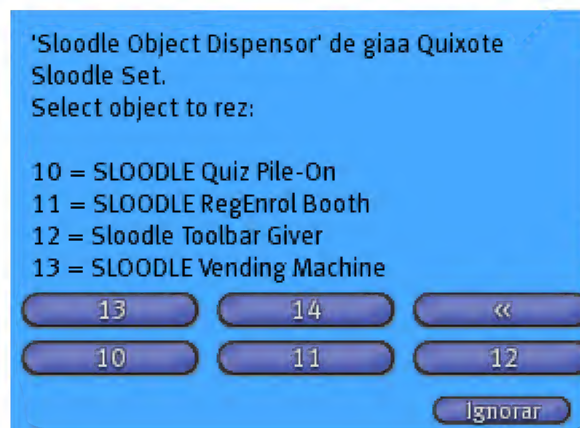
5.19.2. Configuración de RegEnrol_Booth

Vamos a configurar el RegEnrol_Booth.

- Hacemos clic sobre el **rez objects**:



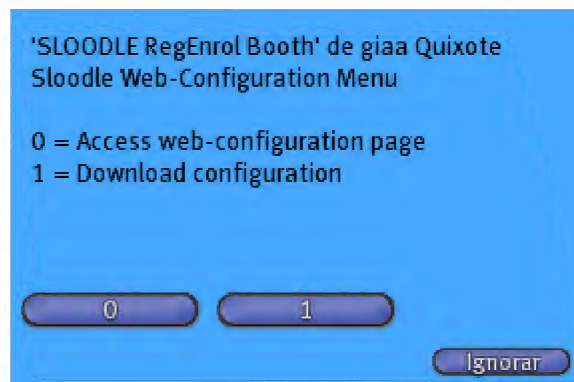
- Nos muestra el siguiente menú:



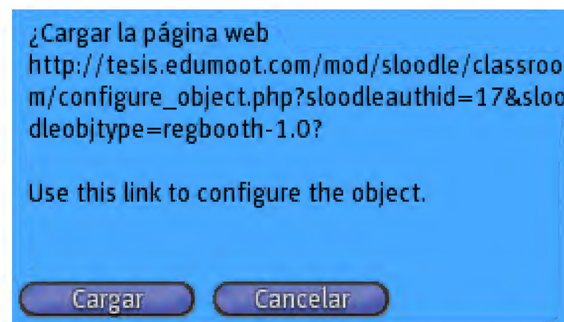
- Hacemos clic en el botón **11** Nos muestra el **RegEnrol_Booth** y hacemos clic en el "RegEnrol_Booth":



- Nos muestra el siguiente mensaje:



- Hacemos clic en el botón **0**. Nos muestra la siguiente página:



- Hacemos clic en el botón **Cargar**. Nos muestra la siguiente página:

Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE RegEnrol Booth
Object UUID: aeb051c2-0604-4520-8223-f43ed19f875d
Object Type: regbooth-1.0

Object Configuration

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Submit

- Hacemos clic en el botón **Submit.** Nos muestra la siguiente página:

Object Configuration

Object Details
Object Name: SLOODLE RegEnrol Booth
Object UUID: aeb051c2-0604-4520-8223-f43ed19f875d
Object Type: regbooth-1.0
Authorized for: OpenSim y OsGrid > SLOODLE
Controller: Controlador para OpenSim

Number of settings stored: 2

Continue

- Hacemos clic en **Continue** y en **RegEnrol Booth**:



- Nos muestra la siguiente pantalla:

'SLOODLE RegEnrol Booth' de giaa Quixote
Sloodle Web-Configuration Menu

0 = Access web-configuration page
1 = Download configuration

0 1

Ignorar

- Hacemos clic en el botón **1**. El Chat nos muestra el siguiente mensaje:

SLOODLE RegEnrol Booth: Configuration received

5.19.3. Prueba de RegEnrol_Booth

5.19.3.1. Prueba de RegEnrol_Booth con un avatar registrado

Vamos a hacer una prueba con RegEnrol_Booth.

- Hacemos clic en el **RegEnrol Booth**:



- El Chat nos muestra el siguiente mensaje:

SLOODLE RegEnrol Booth: Attempting to register and enrol your avatar...
SLOODLE RegEnrol Booth: Thank you giaa Quixote. Your account was already enrolled.

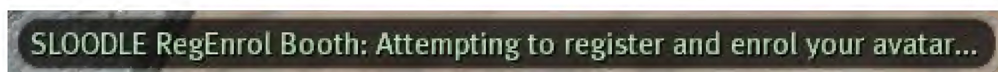
5.19.3.2. Prueba de RegEnrol_Booth con un avatar no registrado

Ahora vamos a utilizamos un avatar (“Saga Bayn”) que no tenga un usuario registrado en nuestro sitio Moodle.

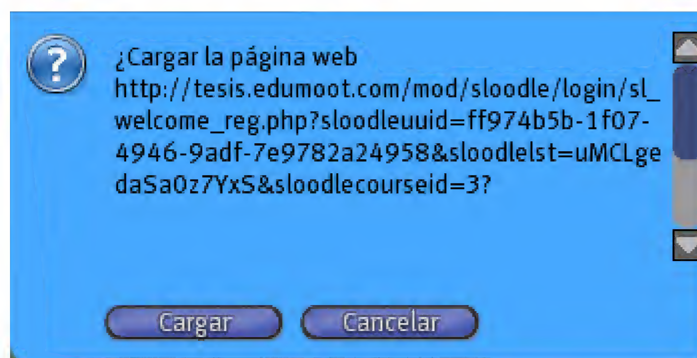
- Hacemos clic en con el botón izquierdo del ratón sobre **RegEnrol_Booth**:



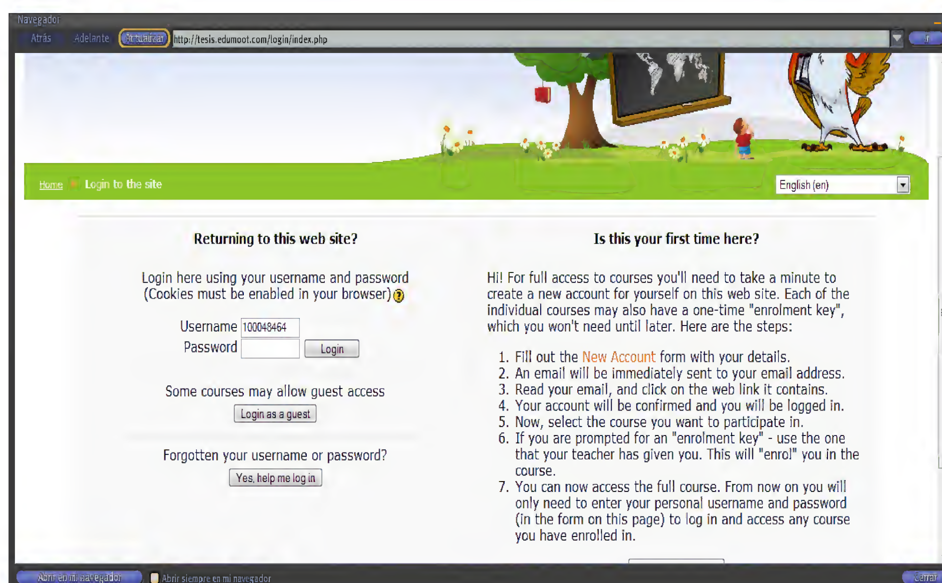
- Nos sale el siguiente mensaje:



- Nos sale la siguiente ventana azul:



- Hacemos clic con el botón izquierdo del ratón el botón **Cargar.** Nos lleva a nuestro sitio Moodle para que hagamos Login:



- Introducimos nuestros datos de usuario de nuestra cuenta Moodle. Hacemos clic en el botón **Login y n**os sale el siguiente mensaje:

Welcome to SLOODLE, saga Bayn

Your avatar was successfully linked to your Moodle account. All SLOODLE objects linked to this site should now recognised you automatically.

- Y depuse nos sale el siguiente mensaje:

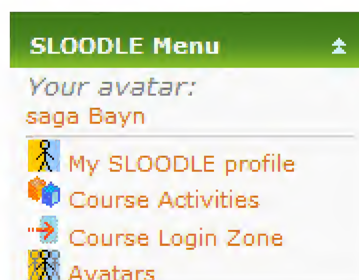
You are about to enrol yourself as a member of this course.
Are you sure you wish to do this?

Yes No

- Hacemos clic en el botón **Yes**.Nos aparece nuestro curso Moodle de **OpenSim**:



- Vamos al Menú **Sloodle Menu**:



- Hacemos clic en el enlace “Avatars”:



Vemos nuestro avatar registra en nuestro sitio Moodle y en nuestra asignatura "OpenSim":

OpenSim y OsGrid

User	Avatar
eduardo rojo	saga Bayn

5.20. Toolbar_Giver

5.20.1. Definición de Toolbar_Giver

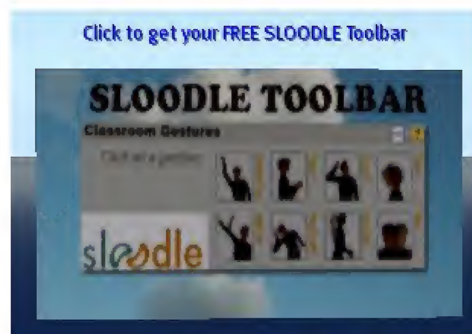
Con esta utilidad, se tienen dos funciones:

- **Gestos del avatar.**

  Permite a nuestro avatar realizar diferentes movimientos (alzar el brazo para realizar una pregunta, aplaudir, negar y confirmar con la cabeza, etc.).

- **Blog.**

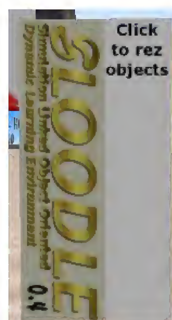
  Permite acceder y escribir en tu blog de la plataforma Moodle desde SL.



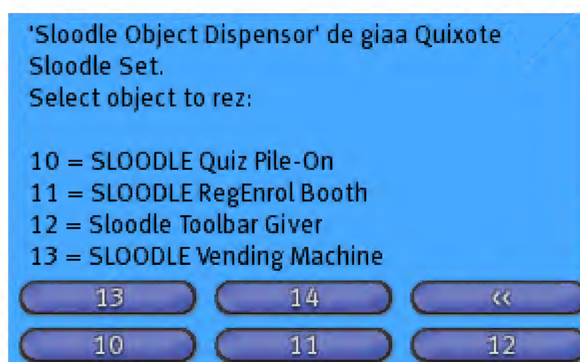
5.20.2. Configuración de Toolbar_Giver

Vamos a configurar la Toolbar_Giver.

- Hacemos clic sobre el **rez objects**:



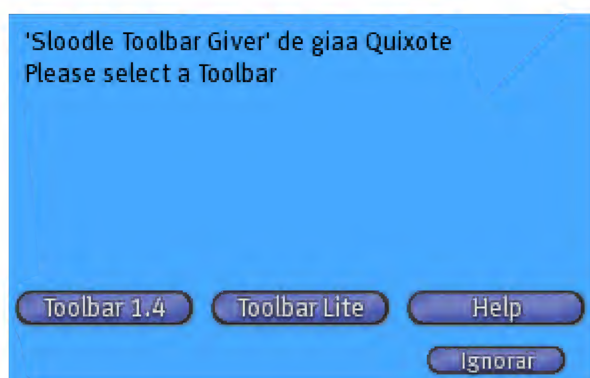
- Nos muestra el siguiente menú:



- Pulsamos el botón 12. Nos muestra el objeto Toolbar y hacemos clic en ella:



- Nos muestra la siguiente ventana:



- Hacemos clic sobre el botón Toolbar 1.4.Nos muestra la siguiente ventana:

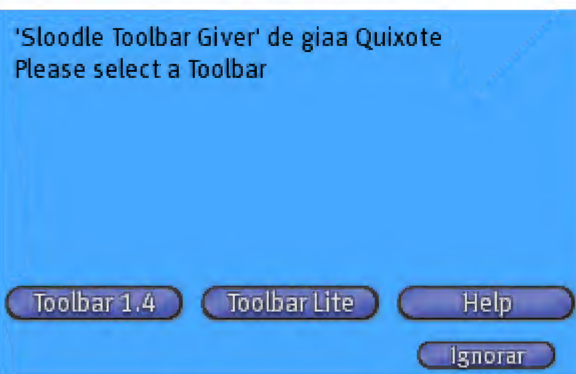
Un objeto llamado Sloodle Toolbar Giver, propiedad de giaa Quixote, le ha dado un/a object de nombre 'Sloodle Toolbar v1.4' .

Guardar Descartar Ignorar

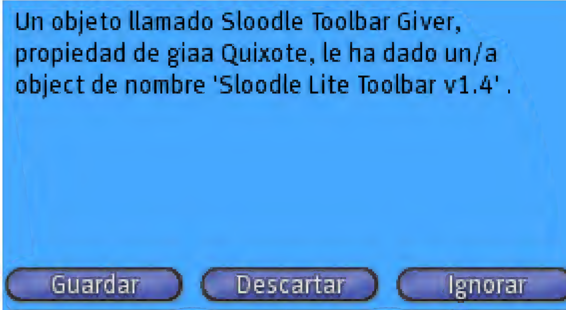
- Hacemos clic en **Guardar**.Presionamos de nuevo en **Toolbar**:



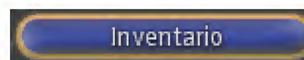
- Nos muestra la siguiente ventana:



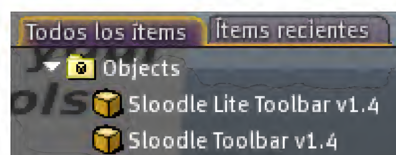
- Hacemos clic sobre el botón **Toolbar Lite.** Nos muestra la siguiente ventana:



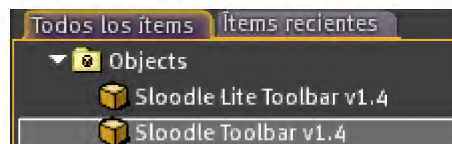
- Hacemos clic en **Guardar**. Comprobamos nuestro inventario:



- En el inventario tenemos 2 objetos nuevos (Toolbar V1.4 y Lite Toolbar v1.4):



- Seleccionamos **Sloodle Toolbar V1.4** desde el inventario:

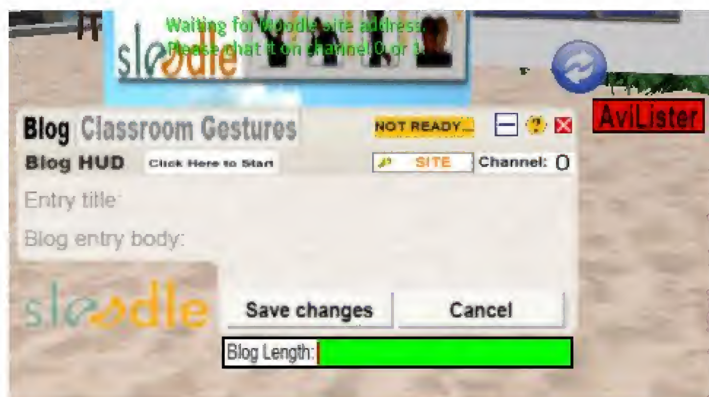


- Podemos activarlo usando la opción "Anexar como HUD", en "Top":

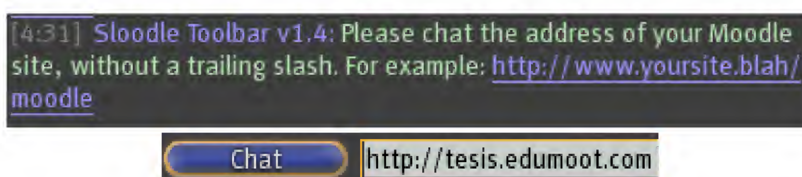


De esta manera será visible en la interfaz de Second Life.

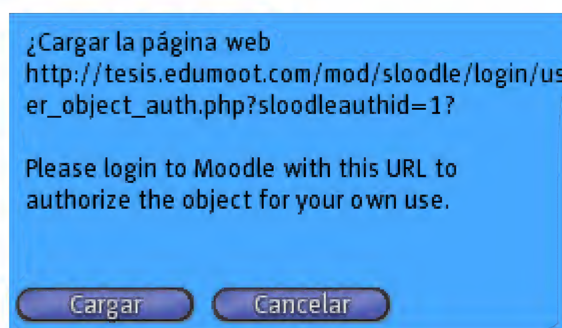
- Al activarse, el Toolbar nos enviara un mensaje privado, preguntando por la ruta del servidor web:



- Escribimos la dirección web en el área Chat:



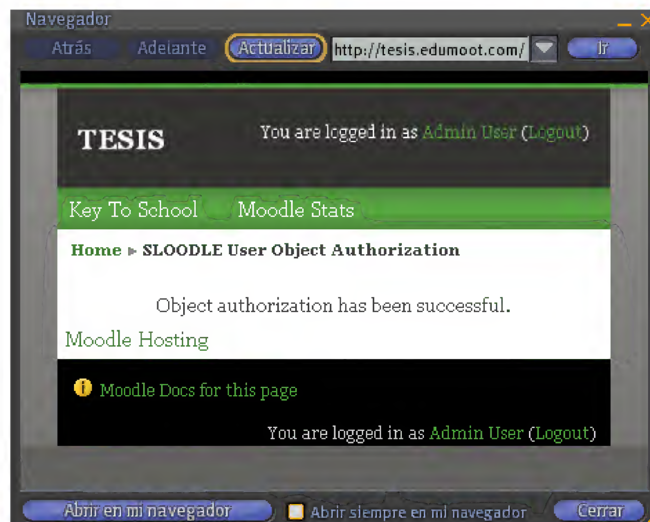
- Nos muestra la siguiente ventana:



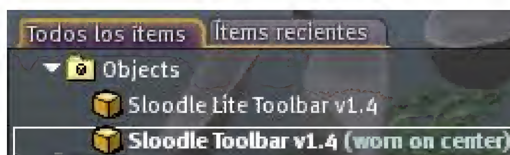
- Hacemos clic en **Cargar**:

A continuación, se abrirá una ventana de navegador, en nuestro servidor Web, donde ingresar nuestro usuario y contraseña de Moodle. La idea aquí es que la conexión actual desde Second Life quede asociada al usuario de Moodle.

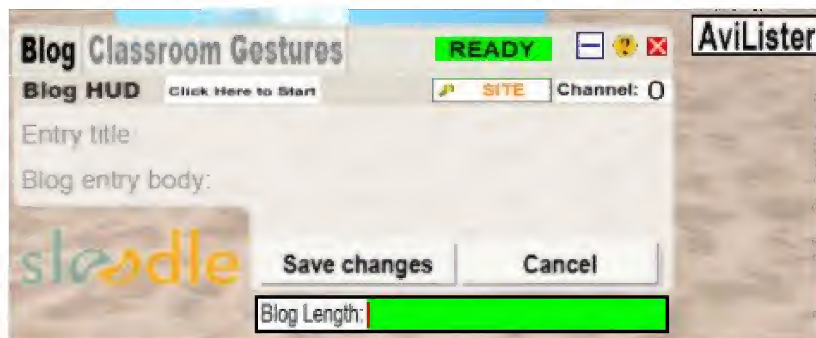
- Nos muestra la siguiente ventana:



- <u>Miramos en el inventario la **Toolbar v1.4**</u>:

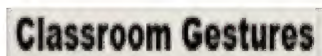


- <u>Y ahora observamos la **Toolbar V1.4**</u>:



5.20.3. Pruebas Toolbar_Giver

5.20.3.1. Seleccionamos en la "Toolbar V1.4", la opción "Classroom Gestures"



Tenemos que hacer "clic en un gesto", para que el avatar lo haga, los gestos que hay son los siguientes:

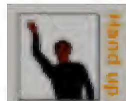
- Hand up: Levantar la mano.
- Wave: Agitar el brazo.
- Clap: Aplaudir.

- Nod off: Dormirse.
- Huh: Rasca su cabeza.
- Got it!:¡Lo tiene!
- Yes: si.
- No.

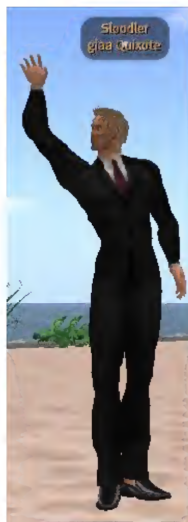
Vamos a comprobar los gestos.

1. **Hand up.**

- Elegimos "Hand up: levantar la mano":



Podemos comprobar como nuestro avatar levanta la mano.

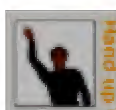


- Si queremos bajar la mano, deberemos de volver a hacer clic en "Hand up":



2. **Yes.**

- Podemos levantar la mano, y a la vez, estar afirmando con la cabeza (afirmar con la cabeza se desconecta automáticamente):





5.20.3.2. Seleccionamos en la Toolbar V1.4, la opción Blog

Vamos a comprobar la opción Blog en la Toolbar.

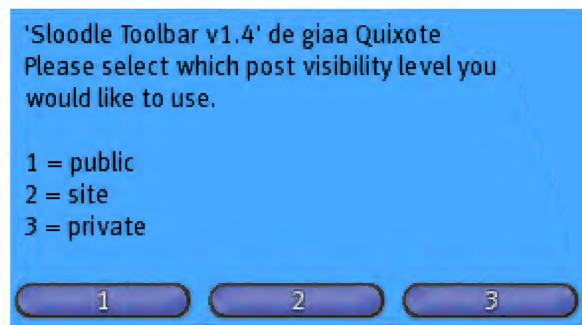
- Tenemos que dar al botón "Clic Here to Start":



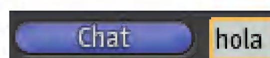
- Hacemos clic en "site":



- Nos muestra la siguiente pantalla:



- Hacemos clic en el botón **1**. Vemos "site" que ha pasado a "public (publico)":
- Escribimos en el Chat "Hola":



- Y ahora observamos la "Toolbar V1.4":



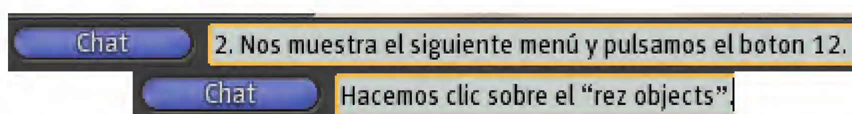
- Escribimos en el Chat:



- Y ahora observamos la "Toolbar V1.4":



- Escribimos en el Chat:



- Hacemos clic en la "Toolbar V1.4" en "Save change". Nos sale el siguiente mensaje en el Chat:

Sloodle Toolbar v1.4: Thank you giaa Quixote. Please wait while your entry is sent...
Sloodle Toolbar v1.4: Updated blog entry successfully.

- Volvemos a hacer de nuevo lo mismo:



- Hacemos clic en la "Toolbar V1.4" en **Save change**. Para verificar que todo va bien, en la ventana de navegador escribimos la dirección:

http://tesis.edumoot.com/blog

- Vemos la página siguiente y se pude apreciar el texto que se escribió en Second Life:

TESIS

Participants | Blogs

Add a new entry

Blog Menu
Add a new entry
View my entries
Blog preferences
View site entries

clase de toolbar
by Admin User - Wednesday, 12 August 2009, 08:01 AM
Anyone in the world
Posted from Second Life: Sloodle vamos a hacer un objeto toolbar Edit | Delete | Permalink

Configuracion de Toolbar
by Admin User - Wednesday, 12 August 2009, 07:42 AM
Anyone in the world
Posted from Second Life: Sloodle como practica quiero que hagais un objeto toolbar. Edit | Delete | Permalink

Practicas
by Admin User - Wednesday, 12 August 2009, 07:19 AM
Anyone in the world
Posted from Second Life: Sloodle Crear un objeto la toolbar Edit | Delete | Permalink

hola
by Admin User - Wednesday, 12 August 2009, 07:16 AM
Anyone on this site
Posted from Second Life: Sloodle Vamos a comenzar la leccion de hoy que trata sobre Toolbar Hacemos clic sobre el "rez objects". 2. Nos muestra el siguiente menú y pulsamos el boton 12. Edit | Delete | Permalink

5.21. Vending Machine

5.21.1. Definición de Vending Machine

Esta utilidad, se usa para distribuir objetos a tus estudiantes.



5.21.2. Configuración de Vending Machine

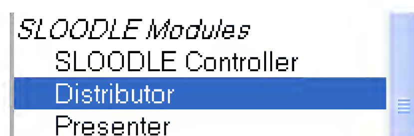
5.21.2.1. Establecer una actividad "distributor" en Moodle

Vamos a establece una actividad distribuidor

- Seleccionamos la "Add an activity":

Add an activity...

- Seleccionamos "Distributor":



- Introducimos en el campo **Name:** Objetos para OpenSim:

Name* Objectos para OpenSim

- Hacemos clic en el botón "Save and return to course":

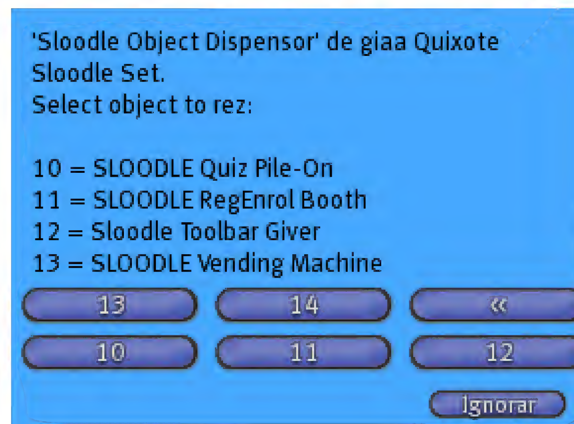
Save and return to course

5.21.2.2. Rez a Vending Machine in Second Life

- Hacemos clic sobre el **rez objects**:



- Nos muestra el siguiente menú:



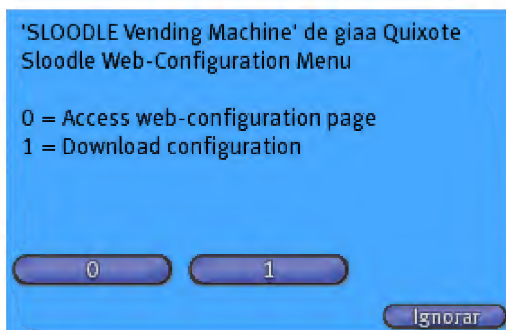
- Hacemos clic en el botón **13**.Nos muestra la "Vending Machine" y hacemos clic en ella:



- La "Vending Machine" le pedirá que escriba su sitio Moodle sin la barra. Haga esto y oprima la tecla Enter:



- Nos muestra el siguiente menú azul, que dice ir a la página:



- Hacemos clic en el botón **0.** Se carga su sitio web Moodle. Si no se ha identificado verá una página de autentificación, entre en su sitio web Moodle:

Returning to this web site?

Login here using your username and password
(Cookies must be enabled in your browser)

Username 100048464
Password *********
Login

- Hacemos clic en **Login.** Seleccione el controlador que desea Conectar:

Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details
Object Name: SLOODLE Vending Machine
Object UUID:
2ea1b4fa-c78f-5fff-7ebd-4451b537082d
Object Type: distributor-1.0

Object Configuration

General Configuration

Select Distributor:

(No Distributor Interface)

Refresh time (seconds): 3600

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Control object: Owner

Submit

Si este objeto se rezzed a través de la configuración de SLOODLE, usted no tendrá que elegir el controlador correcto, después serás automáticamente llevado para el siguiente paso.

- Ahora, seleccione el distribuidor de actividad que desea la máquina que se conecta:

Object Configuration

General Configuration

Select Distributor:

Distributor: Objectos para OpenSim

- Hacemos clic en **Submit**. Nos muestra la siguiente página:

Object Configuration

Object Details

Object Name: SLOODLE Vending Machine

Object UUID: 01c1e826-5652-0480-d88c-fcb71dae0644

Object Type: distributor-1.0

Authorized for: OpenSim y OsGrid > SLOODLE

Controller: Controlador para OpenSim

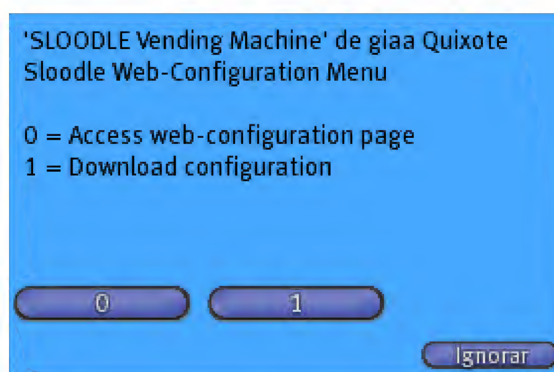
Number of settings stored: 5

Continue

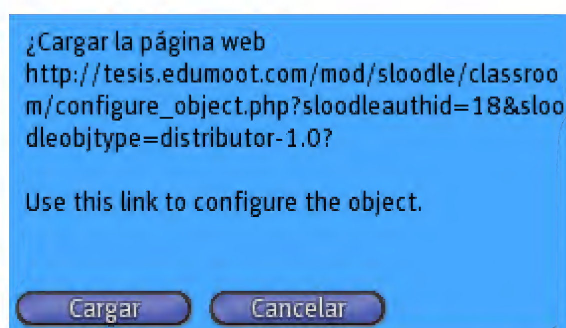
- Hacemos clic en el botón **Continue**. En Second Life, toca la "Vending Machine" de Nuevo:



- Aparecerá un menú azul, selecciona "Download Configuration":



- Hacemos clic en el botón **1**. Nos muestra la siguiente ventana:



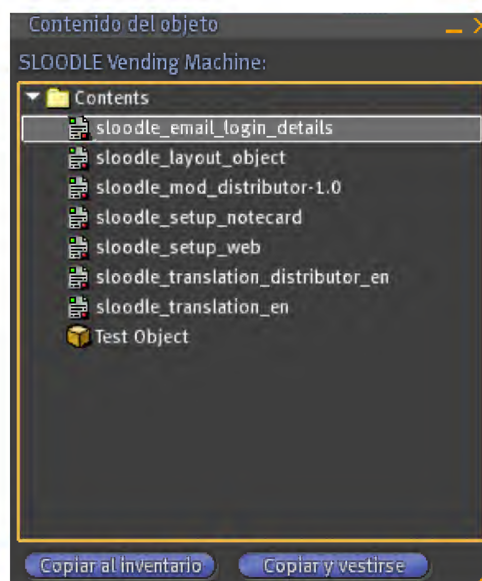
- Hacemos clic en el botón **Cargar**. Recibimos un mensaje en el Chat:

SLOODLE Vending Machine: Configuration received
SLOODLE Vending Machine: Connected successfully

- Tu Vending Machine esta ahora para usarla. Hacemos clic con el botón derecho en “Abrir”:



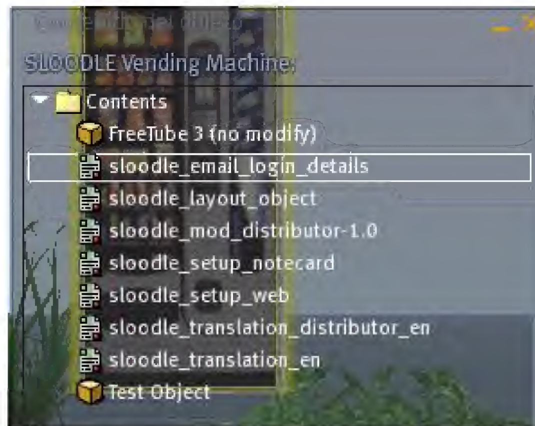
- Nos aparece el siguiente menú:



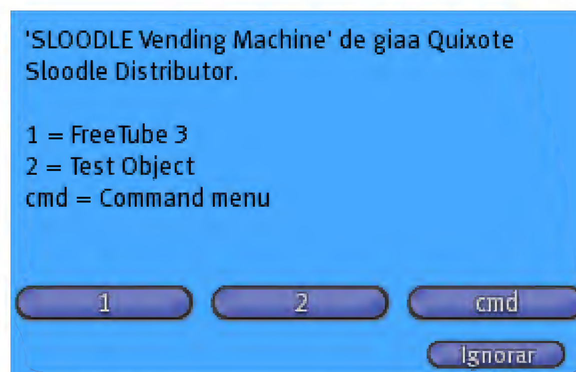
- Abrimos nuestro inventario. Seleccionamos un objeto del inventario:



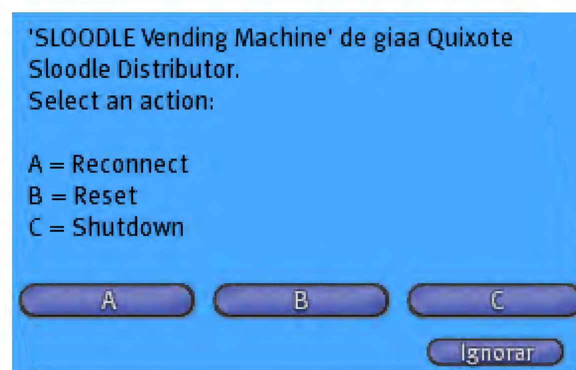
- Lo arrastramos “contenido” de la Vending Machine:



- Cuando hayas terminado de arrastrar objetos, toca de nuevo la “Vending Machine”. Un menú azul aparecerá, en el menú selecciona “cmd”:



- Hacemos clic en el botón “cmd”. Selecciona “Reconnect”:



Esto recargara los detalles de los objetos que tu tienes puesto dentro de tu “Vending Machine”, así que ellos pueden ser vistos en tu Moodle Website.

- Hacemos clic en el botón **A**. Ahora vamos a nuestro sitio Moodle, y hacemos clic en el “Distribuitor: Objetos para OpenSim”:

Distributor: Objectos para OpenSim.

- Nos aparece una nueva ventana, en la cual podemos apreciar que los objetos seleccionados de nuestro avatar se ha actualizado en Moodle, apareciendo "FreeTube3".

Send to another avatar

Select User: giaa Quixote

Select Object: FreeTube 3

FreeTube 3

Send to an Test Object

5.21.3. Prueba de Vending Machine

Vamos a probar la Vending Machine.

- <u>Volvemos a Second Life y hacemos clic en la "Vending Machine"</u>:



- <u>Nos aparece un menú</u>:

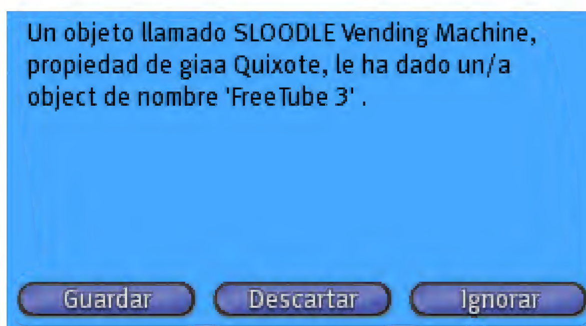
'SLOODLE Vending Machine' de giaa Quixote
Sloodle Distributor.

1 = FreeTube 3
2 = Test Object
cmd = Command menu

1 2 cmd

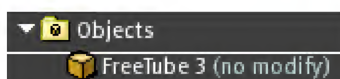
Ignorar

- Hacemos clic en el botón **1**.Nos aparece el siguiente menú azul:



- Hacemos clic en **Guardar**.Hacemos clic en el inventario:

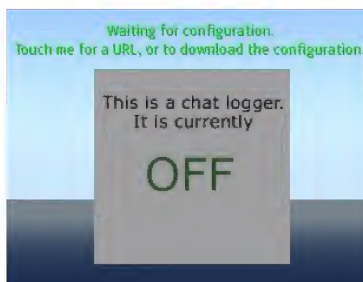
- Y vemos tenemos un objeto que se llama "FreeTube 3" que nos lo ha dado la "Vending Machine":



5.22. WebIntercom

5.22.1. Definición de WebIntercom

Con esta utilidad, podemos interconecta nuestro chat de Moodle con el chat de Second Life.



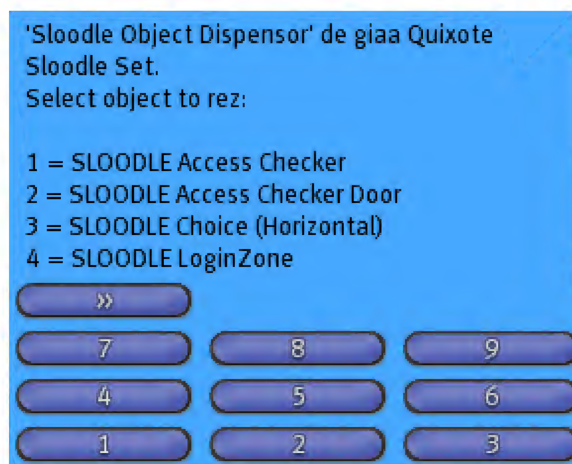
5.22.2. Configuración del WebInterCom

Vamos a configurar el WebInterCom.

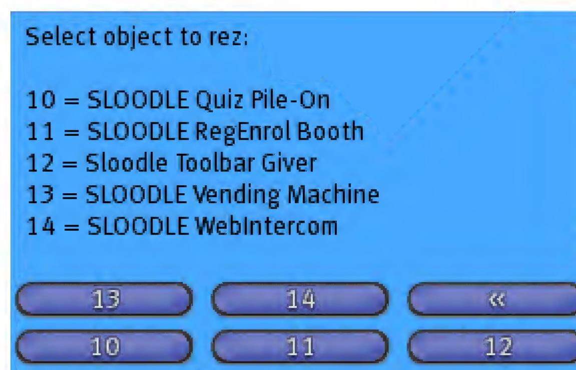
- Ahora pulsamos sobre el bloque para "desplegar (rez)" objetos:



- Nos sale el siguiente menú azul:



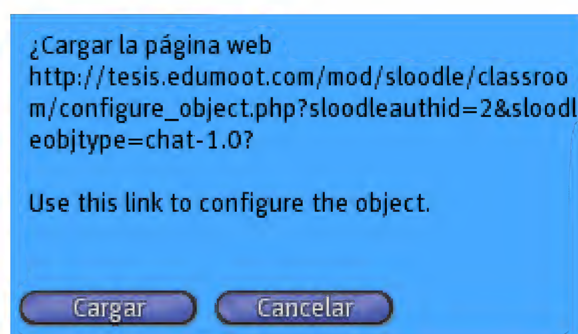
- Buscamos el "SLOODLE WebIntercom", que es el botón **14**:



- Como vemos se a creado, y para activarlo (pues esta a OFF) hay que hacer clic sobre él:



- Nos sale la siguiente menú azul:



Nos indica que hay que usar el link para configurar el objeto, con lo cual hay que dar al botón "Cargar".

- Hay que hacer clic en el botón **Cargar**. Nos carga la página web:

Home ▸ SLOODLE Module ▸ Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Details

Object Name: SLOODLE WebIntercom

Object UUID: 2330f6a4-6d95-baf5-bf28-5546484a4c84

Object Type: chat-1.0

Object Configuration

General Configuration

Select Chatroom:

Chat principal de OS y OG

Listen to object chat: No

Allow auto-deactivation: Sí

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Control object: Owner

Server Access Level

This determines who may use the server resource

Access Level: Public

Submit

Nos indica que el objeto está autorizado

- Tenemos que configurar el "WebIntercom":

- **Selecciono el Chat.**

  Seleccionamos el chat al cual quiero vincular este “WebIntercom”.

Select Chatroom:
Chat Tema 1. Descargas y carpetas

- **Use object (uso del objeto).**

  Lo dejamos público, para que lo pueda usar cualquiera.

- **Control object (control del objeto).**

  Lo dejamos en Owner, para que sea el propietario el que lo controle.

- **Access Level (nivel de acceso)**.

  Lo dejamos en público.

- Una vez que configuremos el objeto como deseemos, pulsamos sobre **Submit**:

- Nos sale el siguiente mensaje:

Home ▸ SLOODLE Module ▸ Object Configuration

Object Details
Object Name: SLOODLE WebIntercom
Object UUID: 2330f6a4-6d95-baf5-bf28-5546484a4c84
Object Type: chat-1.0
Authorized for: OpenSim y OsGrid > SLOODLE
Controller: Controlador descargas y carpetas

Number of settings stored: 7

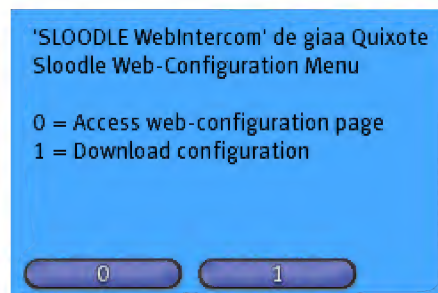
Continuar

Moodle Hosting

- Y damos al botón **Continuar**. Volvemos a pulsar sobre el bloque del WebIntercom:



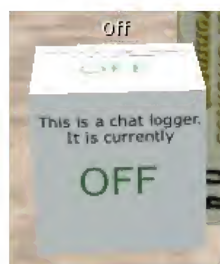
- Y nos sale el siguiente cuadro de texto:



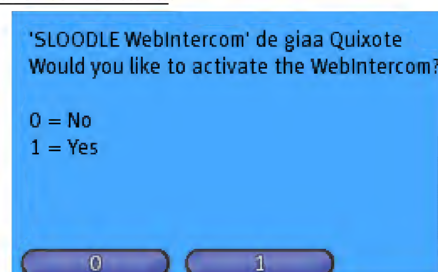
- Elegimos **1** porque queremos descargar la configuración que acabamos de configurar. Ahora tiene que indicar "Configuration received (la configuración ha sido recibida)":



- Nuestro objeto ya está preparado, ahora hacemos clic sobre el objeto:

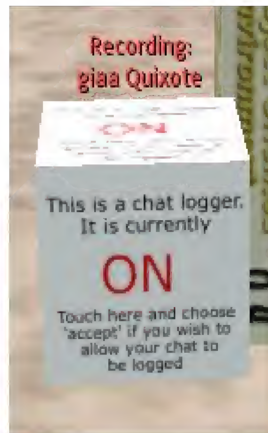


- Y nos muestra la siguiente ventana:



Nos indica que si queremos activar el WebIntercom.

- Pulsamos sobre el botón **1.** Ya esta creada la conexión con Moodle:



SLOODLE WebIntercom: Chat logging is on!
SLOODLE WebIntercom: Join this Moodle chat at http://tesis.edumoot.com/mod/chat/view.php?id=13
SLOODLE WebIntercom: Touch logger to record your chat

5.22.3. Pruebas de WebInterCom

Vamos a hacer pruebas con la herramienta WebInterCom.

- Entramos en nuestro curso mediante el explorador web:

http://tesis.edumoot.com

Curso de OpenSim y OsGrid

- Y entramos en el Chat que está conectado con sloodle:

Chat Tema 1. Descargas y carpetas

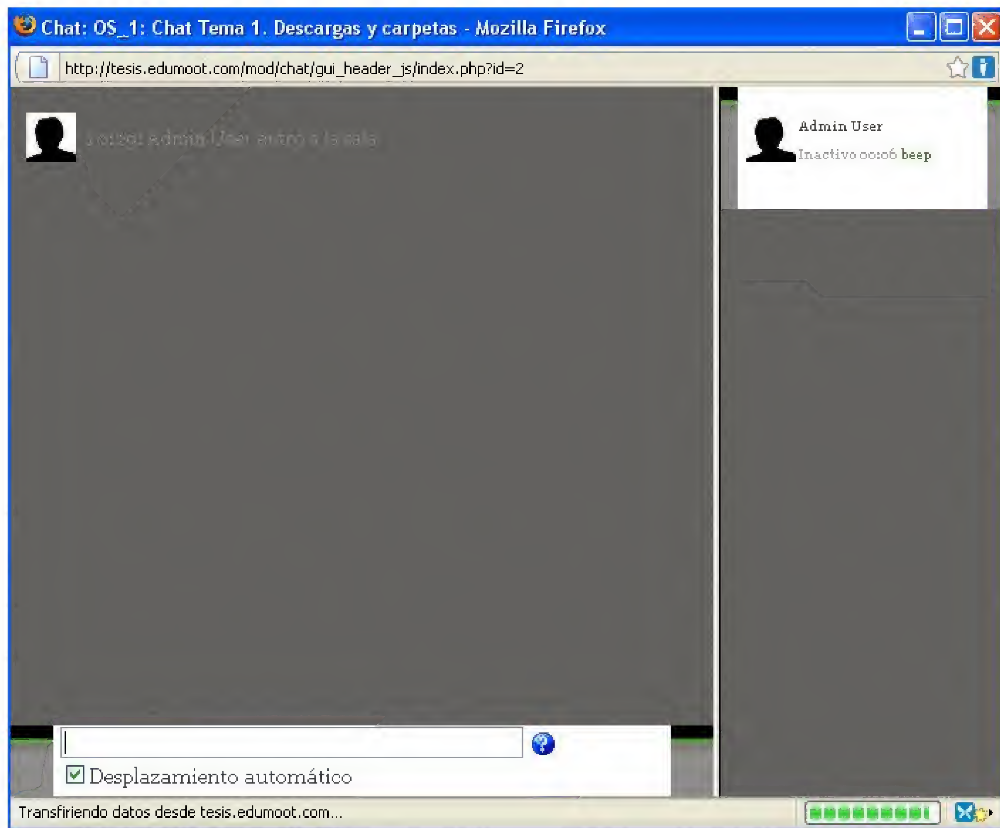
- Ahora entramos en la sala:

Chat Tema 1. Descargas y carpetas

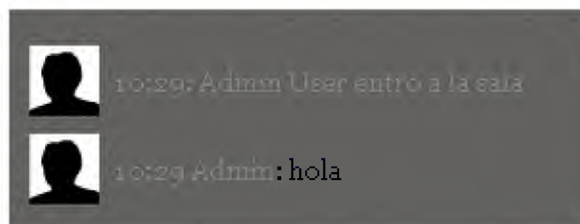
Entrar a la sala

(Versión sin marcos ni JavaScript)

- Nos sale la siguiente ventana:



- Enviamos un mensaje desde Moodle:



- Miramos ahora en Second Life y nos sale el mensaje que hemos enviado desde Moodle:



Conclusiones Sloodle

Sloodle es una estupenda herramienta para dar clase en Second Life, tiene la ventaja de que es gratuita, y además están continuamente sacando nuevas versiones con nuevas herramientas para dar clase.

Sloodle basa sus herramientas de dar clase en seriedad para poder permitir al profesor dar las clases de forma perfecta, como por ejemplo, controlar los movimientos del avatar que maneje el profesor para hacer gestos y de esta forma permitir que los alumnos entiendan las explicaciones del profesor lo mejor posible.

Y Sloodle también basa sus herramientas de dar clase en diversión, para que las explicaciones del profesor se quede en la mente de los alumnos, y también vale para que los alumnos al ver que es divertido participar en las actividades de clase se animen más a participar, un ejemplo, es la herramienta Quiz Pile-On, que son unos pequeños asientos suspendidos en el cielo, en los cuales los alumnos se sientan, entonces a los alumnos sentados en los asientos les sale unas preguntas, y si alguno contesta mal a la pregunta el asiento cae con el avatar del alumno, es una manera divertida de hacer un cuestionario.

CAPÍTULO VI

BOT CONVERSACIONAL

Capítulo 6

Bot Conversacional

6.1. Introducción

Un bot (acortamiento de robot) es un programa informático que realiza funciones muy diversas, imitando el comportamiento de un humano. En este capítulo del Proyecto Final de Carrera explicamos:

- Cómo crear un bot mediante la librería OpenMetaverse.
- Cómo controlar un bot mediante la utilización de la voz, empleando técnicas de Procesamiento de Lenguaje Natural. Un avatar (Master) le da una orden al bot, y éste la acata. Justamente en esto se basa un bot (robot), en recibir órdenes de alguna persona y hacer que las acate.
- Cómo hacer que un bot responda a preguntas dentro de una determinada tarea, mediante un análisis automático de las mismas mediante gramáticas y la generación de respuestas detalladas mediante el uso de bases de datos.

6.2. Creación de nuestro primer bot

6.2.1. Instalación de la librería LibOpenMetaverse

La librería LibOpenMetaverse es una librería .Net basada en Cliente/servidor, usada para acceder y crear mundos virtuales en 3D.

- Descargamos la librería "LibOpenMetaverse" en la siguiente dirección Web:

  http://www.openmetaverse.org/projects/libopenmetaverse

- La biblioteca consta de varios .NET:

- **OpenMetaverse.dll.**

  El núcleo de la biblioteca OpenMetaverse, que puede ser usado para crear aplicaciones clientes y servidor.

- **OpenMetaverseTypes.dll.**
  Funciones específicas 3D y bibliotecas matemáticas.

- **OpenMetaverse.StructuredData.dll.**
  Un conjunto de librerías para dar un completo soporte a (Linden Lab Structured Data (LLSD) y JSON (JavaScript Object Notation).

- **OpenJpeg.**
  Un envoltorio .NET para codificar y descodificar los datos JPEG2000.

La librería OpenMetaverse es compatible con Microsoft .NET Framework 2.0 y versiones posteriores, así como Mono 1.9 y posterior.

- Para más detalles usa la wiki de los desarrolladores:

  http://lib.openmetaverse.org/wiki/Main_Page

- Vamos a la última realización de la "LibOpenMetarverse", en este caso es la realización 0.7.0:

LibOpenMetaverse Release 0.7.0

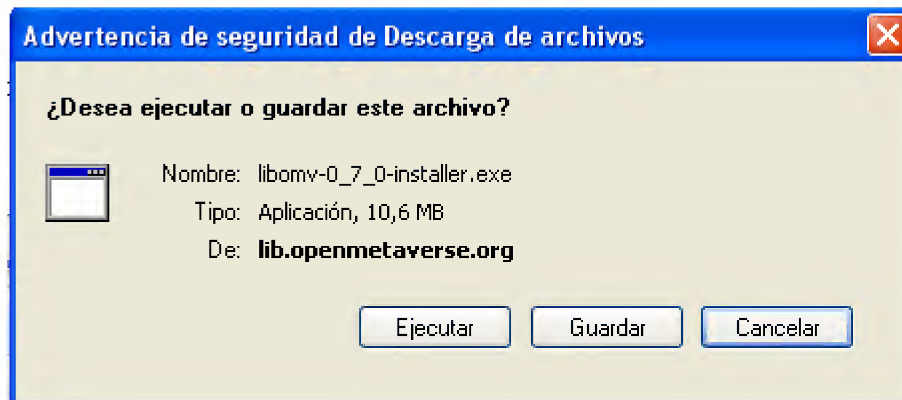
- Ahora vamos al enlace de descarga:

  http://openmetaverse.org/projects/libopenmetaverse/download

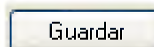
- Ahora hacemos clic en el enlace que pone:

  Downlocad a precompiled binary versión with Windows installer

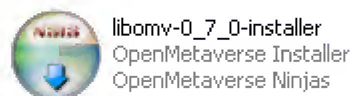
- Nos muestra el siguiente ventana emergente:



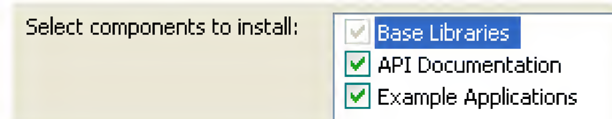
- Hacemos clic en "Guardar":



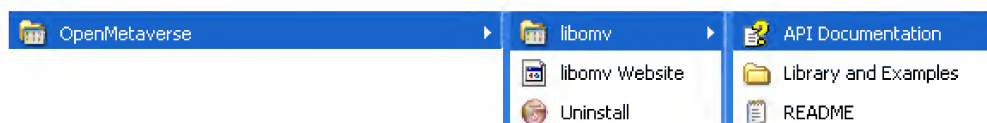
- Tenemos el siguiente instalador:



- Hacemos clic en el instalador para instalarlo, en la instalación es mejor dejar que se instale también la API de OpenMetaverse:



- Nos creara la siguiente carpeta:



Ruta: C: //archivos de programa/ OpenMetaverse / libomv / bin

6.2.2. Crear un proyecto para tu bot en Visual Studio

Vamos a crear un proyecto para el bot en Visual Studio.

- Abre Visual Studio, Selecciona **archivo → nuevo → proyecto**:



- Selecciona **Visual C# → Windows**:



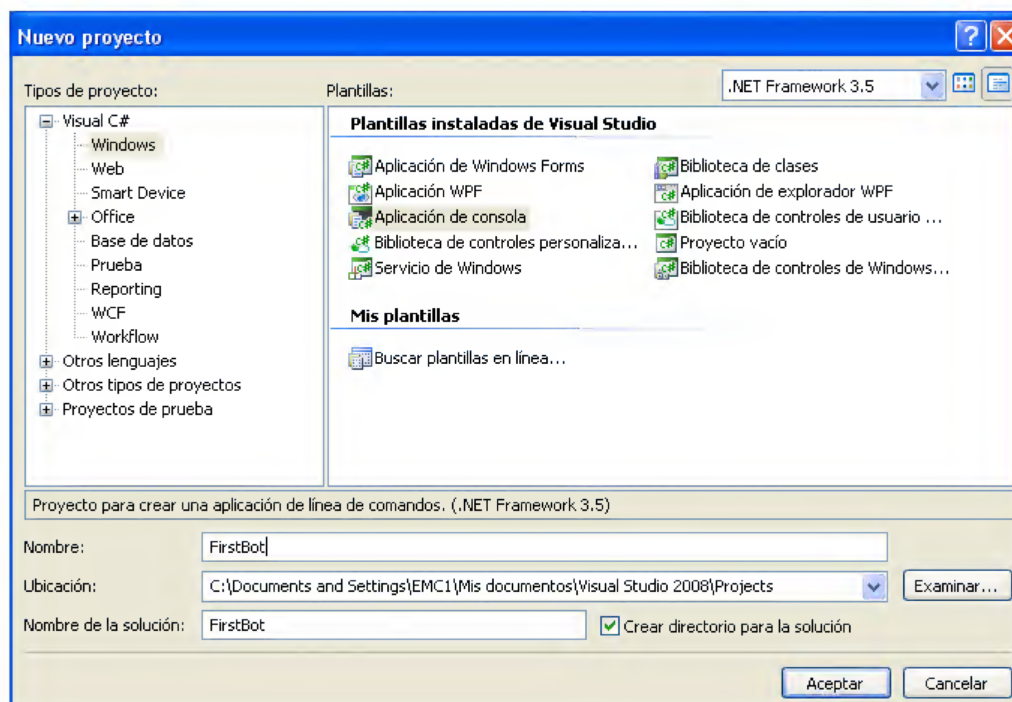
- Selecciona **Aplicación de consola** desde la lista de plantillas:



6.2.2.1. El nombre del proyecto y la ubicación

Vamos a dar nombre al proyecto y ubicación.

- Nos aparece la siguiente pantalla:



- **Nombre.**

Nombre: FirstBot

- **Ubicación.**

Ubicación: C:\Documents and Settings\EMC1\Mis documentos\Visual Studio 2008\Projects

Selecciona una localización en tu disco duro.

- **Nombre de la solución**.

➢ **Nombre:** FirstBot.

➢ Chequea "**Crear directorio para la solución**".

Crear directorio para la solución

- Clic en el botón Aceptar:

Aceptar

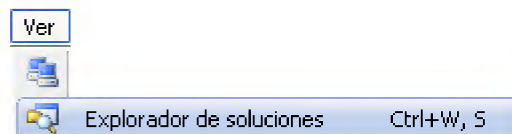
- Nos aparece la siguiente pantalla:



6.2.2.2. Añadimos las requeridas referencias a las librerías a tu proyecto

Vamos a añadir las referencias a las librerías.

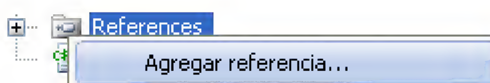
- Haz clic en **Ver → Explorador de soluciones** en la barra de herramientas:



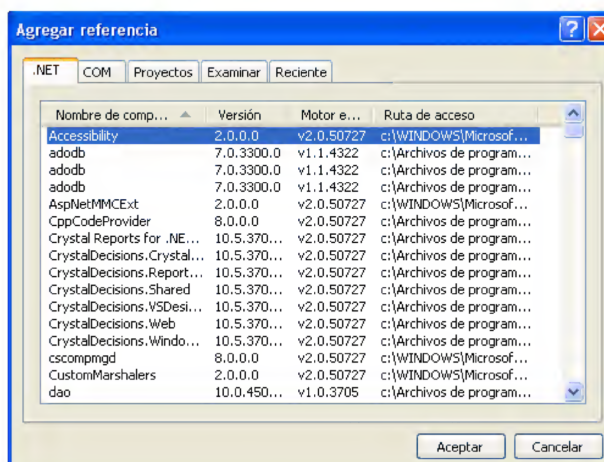
- Nos muestra la siguiente pantalla:



➢ En el **Explorador de soluciones** haga clic con el botón derecho del ratón en el **References**, entonces haz clic en **Agregar referencia**:



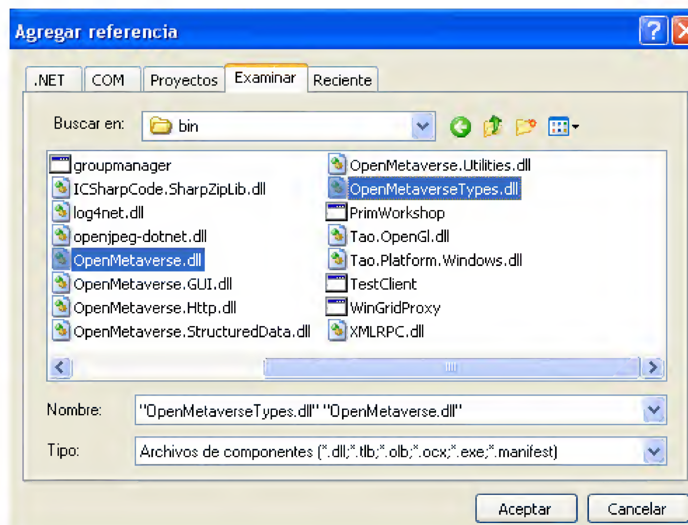
➢ <u>Nos muestra la siguiente pantalla</u>:



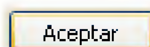
➢ En el cuadro de diálogo **Agregar referencia** haga clic en la ficha **Examinar**, ahora navega al directorio donde tu compilaste la "libOpenMetaverse library" (tu deberías ver una carpeta llamada "bin", ábrela).

Ruta: C: //archivos de programa/ OpenMetaverse / libomv / bin

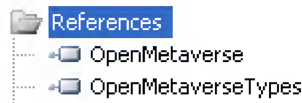
➢ <u>Ahora selecciona las siguientes filas (para seleccionar varios archivos mantén pulsada la tecla "Ctrl" en el teclado antes de hacer clic en cada uno de los archivos)</u>: OpenMetaverse.dll y OpenMetaverseType.dll.



➢ <u>Haga clic en el botón "Aceptar"</u>:



➢ Nos aparecen la librería "OpenMetaverse" y "OpenMetaverseType":



➢ Si usted recibe una advertencia de diálogo preguntando por la dependencia versiones, haga clic en el botón "Apply all ítems (Aplicar a todos los artículos)", a continuación, haga clic en el botón Sí.

➢ Ahora copia el siguiente código en el editor (debe ser por defecto Program.cs) (**Nota:** Tú tienes que cambiar los valores para el nombre avatar, el apellido y el Password).

6.2.2.3. Código fuente del primer Bot

Vamos a ver el código fuente del primer Bot.

- Copiamos el siguiente código fuente en nuestro proyecto:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;

namespace bot2
 {
    class Program
    {
        public static GridClient Client = new GridClient();
        private static string first_name = "giaa";
        private static string last_name = "Quixote";
        private static string password = "*************";

       static void Main(string[] args)
       {
         Client.Network.OnConnected += new
          NetworkManager.ConnectedCallback(Network_OnConnected);


         if (Client.Network.Login(first_name, last_name, password, "My First
         Bot", "Your name"))
          {
            Client.Appearance.SetPreviousAppearance(true);
            Console.WriteLine("I logged into Second Life!");
          }
         else
```

```
        {
            Console.WriteLine("I couldn't log in, here is why:
            Client.Network.LoginMessage);
            Console.WriteLine("press enter to close...");
            Console.ReadLine();
        }
    }

    static void Network_OnConnected(object sender)
    {
            Console.WriteLine("I'm connected to the simulator, going to greet
            everyone around me");
            Client.Self.Chat("Hello World!", 0, ChatType.Normal);
            Console.ReadLine();
            Console.WriteLine("Now I am going to logout of SL.. Goodbye!");
            Client.Network.Logout();
            Console.WriteLine("I am Loged out please press enter to close...");
            Console.ReadLine();
    }
  }
}
```

- **Las referencias.**

```
using OpenMetaverse;
```

Esto le indica al compilador que incluya el contenido de las librerías OpenMetaverse cuando compile este archivo, pon esto en lo alto de todos los ficheros que usen algo de la clase OpenMetaverse.

- **La clase GridClient.**

```
public static GridClient Client = new GridClient();
```

Esta línea es probablemente la línea más importante en cualquier aplicación OpenMetaverse. Ella contiene todas las funciones que son requeridas para que su bot se comunique con el grid de Second Life.

- **Definiendo algunas variables.**

```
private static string first_name = "giaa";
private static string last_name = "Quixote";
private static string password = "*************";
```

Este paso es opcional, pero es incluido por claridad. Básicamente se separa el nombre y el apellido del bot en dos variables y también se pone la contraseña en una variable.

Tenga en cuenta que este método no es seguro en absoluto, y nunca debe incluir los detalles de una cuenta de SL en el código si quieres distribuir tu aplicación. Ello está bien si usted es el único que va a utilizar la aplicación.

- **El evento Main.**

Este bloque de código se ejecuta cuando se inicia el programa.

Tenga en cuenta que LibOpenMetaverse utiliza hilos, por lo que el programa actual no termina hasta que logees tu bot fuera de Second Life.

- **El evento OnConected.**

```
Client.Network.OnConnected += new
 NetworkManager.ConnectedCallback(Network_OnConnected);
```

El evento OnConnected es definido aquí. Básicamente, cada vez que el robot se conecta a un simulador y las marca su presencia allí, se disparará el método Network_OnConnected (que se define a continuación el bloque principal).

- **Logging In.**

```
if (Client.Network.Login(first_name, last_name, password, "My First Bot", "Your
name"))
```

Hemos definido la variable de Second Life ("cliente") y define lo que el bot debe hacer cuando se conecta (OnConnected). Ahora es el momento de registrar el bot en el grid.

- El método **Login**() devuelve un valor boolean: True si aparece un éxito y falso si aparece un error.
- Reemplace "**Your Name**" ya sea con el nombre de su vida real, su nombre de SL, o su dirección de correo electrónico. Esto se envía a LL como extra en el inicio de sesión.
- Reemplace "**My First Bot**" con el nombre de su aplicación. Esto se envía a LL como extra en el inicio de sesión.

- **Error?.**

```
Console.WriteLine("I couldn't log in, here is why: " +
 Client.Network.LoginMessage);
```

Una propiedad muy útil en la clase de red es “LoginMessage”. Por un error, este string se llena con los detalles del error. Si no hubo errores al iniciar la sesión, el LoginMessage contendrá el mensaje del día.

- **Chat.**

```
Client.Self.Chat("Hello World!", 0, ChatType.Normal);
```

En el método Network_OnConnected, se ejecuta esta línea de código. Éstos son los parámetros:

- El mensaje (string).
- El canal (integer) (usa el 0 para el uso público chat).
- El tipo de chat (Enum) .

Los tipos de chat más comúnmente utilizados son los siguientes:

- ChatType.Normal = Char normal, 20 metros de rango.
- ChatType.Whisper = Chat susurro, 10 metros rango.
- ChatType.Shout = Chat gritar, 100 metros rango.

- **Logging out.**

```
Client.Network.Logout();
```

Ahora que hemos terminado le permitira salir.

6.2.2.4. Compilar y ejecutar la aplicación

Vamos a compilar y ejecutar la aplicación.

- Entramos a Second Life con “Saga Bayn” que es otro avatar.
- Presiona tu tecla F5 para compilar y ejecutar tu aplicación FirstBot .
- Si todo ha ido bien debería haber un mensaje que dice:

“You have successfully logged into Second Life!, The Message of the day is {0}, Press any Key to Logout

Es decir, correctamente conectado, el mensaje del día es 0, presiona cualquier tecla para desconectarte.

- <u>En otro caso</u>:

  Usted debe obtener una indicación de que porque no te has conectado.

- <u>Vemos la consola que se ha abierto</u>:

```
file:///C:/Documents and Settings/EMC1/Mis documentos/Visual Studio 2008/Projects/FirstB...
125 [XML-RPC Login] INFO  - No log configuration found, defaulting to console lo
gging
203 [XML-RPC Login] INFO  - <giaa Quixote>: Connecting to (216.82.38.187:12035)
890 [16] INFO  - <giaa Quixote>: Received a region handshake for Sloodle (216.82
.38.187:12035)
You have successfully logged into Second Life!
 The Message of the day is What happened with Second Life's economy during the l
ast quarter? CLICK HERE to the detailed report: http://bit.ly/sleq209
Press any Key to Logout
7375 [15] DEBUG - Capabilities event queue connected
```

- <u>Vemos que el bot "giaa Quixote" nos saluda en el Chat de Second Life</u>:

giaa Quixote: Hello World!

- <u>Comprobamos que nuestro bot "giaa Quixote" es una nube</u>:



- <u>Que el bot sea una nube es debido a que no hemos incluido la siguiente línea de código</u>:

  Client.Appearance.SetPreviousAppearance(true);

- <u>Ahora presionamos cualquier tecla y el bot giaa Quixote se desconecta de Second Life</u>:

```
179921 [9] INFO  - <giaa Quixote>: Logging out
180125 [16] INFO  - <giaa Quixote>: NetworkManager shutdown initiated
180125 [16] INFO  - <giaa Quixote>: Caps system for Sloodle (216.82.38.187:12035
) is aborting
Press Any Key to Exit
```

6.3. Códigos fuentes para el bot

Vamos a ver el código fuente de un bot que se registra en la Grid SL y dice “Hola” cuando algún avatar le dice “Hola” y “Adiós” cuando alguien le dice “Adiós”.

- Código fuente del bot:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;

namespace Bot3
{
class Program
   {
      public static GridClient Client = new GridClient();
      private static string first_name = “giaa”;
      private static string last_name = “Quixote”;
      private static string password = “***********”;

      static void Main(string[] args)
      {
            Client.Network.OnConnected += new
            NetworkManager.ConnectedCallback(Network_OnConnected);

            if (Client.Network.Login(first_name, last_name, password, “Program”,
            “e100048464@gmail.com”))
            {
               Client.Appearance.SetPreviousAppearance(true);
               Console.WriteLine(“Yo estoy conectado a Second Life!”);
               Client.Self.OnChat += new
               AgentManager.ChatCallback(Self_OnChat);
            }
            else
            {
              Console.WriteLine(“Yo no puedo conectarme aqui, es porque: “ +
              Client.Network.LoginMessage);
              Console.WriteLine(“Presione cualqueir tecla para cerrar...”);
              Console.ReadLine();
            }
       }

      static void Network_OnConnected(object sender)
      {
        Console.WriteLine(“Yo estoy conectado al simulador, Yo voy a saludar a
```

```
    todo el mundo alrededor mio");
    Client.Self.Chat("Hola a todo el mundo!", 0, ChatType.Normal);
    Console.ReadLine();
    Console.WriteLine("Ahora yo voy a desconectarme de SL.. Adios!");
    Client.Network.Logout();
    Console.WriteLine("Yo estoy desconectado, por favor presione enter
    para cerrar...");
    Console.ReadLine();
  }


  static void Self_OnChat(string message, ChatAudibleLevel audible, ChatType type,
  ChatSourceType sourceType, string fromName, UUID id, UUID ownerid, Vector3
  position)
  {
    string mensaje_minuscula = message.ToLower();
    UUID typing = new UUID("c541c47f-e0c0-058b-ad1a-d6ae3a4584d9");

    if (mensaje_minuscula.Equals("hola"))
    {
      Client.Self.AnimationStart(typing, true);
      Client.Self.Chat("Hola Sr. " + fromName, 0, ChatType.Normal);
      Client.Self.AnimationStop(typing, true);
    }
    else if (mensaje_minuscula.Equals("adiós"))
    {
      Client.Self.AnimationStart(typing, true);
      Client.Self.Chat("Adiós Sr. " + fromName, 0, ChatType.Normal);
      Client.Self.AnimationStop(typing, true);
     }
    }
  }
}
```

6.3.1. Explicación del código fuente

Vamos a explicar el código fuente.

- La siguiente línea de código vale para que se active la apariencia del bot y así no se vea como una nube:

```
Client.Appearance.SetPreviousAppearance(true);
```

- La siguiente línea de código vale para conectar el chat del bot:

```
Client.Self.OnChat += new AgentManager.ChatCallback(Self_OnChat);
```

- La siguiente línea de código vale para identificar la animación que queremos cargar:

```
UUID typing = new UUID("c541c47f-e0c0-058b-ad1a-d6ae3a4584d9");
```

- Con la siguiente línea de código respondemos a las preguntas que nos formulen en el chat:

```
static void Self_OnChat(string message, ChatAudibleLevel audible, ChatType type,
ChatSourceType sourceType, string fromName, UUID id, UUID ownerid, Vector3
position)
 {
   string mensaje_minuscula = message.ToLower();

   if (mensaje_minuscula.Equals("hola"))
   {
    Client.Self.Chat("Hola Sr. " + fromName, 0, ChatType.Normal);
   }
   else if (mensaje_minuscula.Equals("adiós"))
   {
    Client.Self.Chat("Adiós Sr. " + fromName, 0, ChatType.Normal);
   }
 }
```

- Éstas líneas de código se utilizan para activar y parar la animación de escribir a máquina cuando se escribe en el chat:

```
Client.Self.AnimationStart(typing, true);
 Client.Self.AnimationStop(typing, true);
```

6.3.2. Compilar y ejecutar la aplicación

Vamos a compilar y ejecutar la aplicación.

- Presiona tu tecla F5 para compilar y ejecutar tu aplicación.

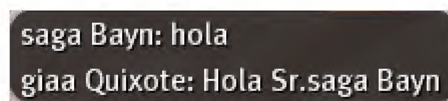
- Vemos al bot "giaa Quixote" conectándose a Second Life:



- Le decimos “hola” por el Chat:



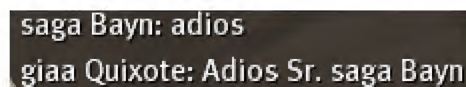
- El bot nos responde “Hola Sr. Saga Bayn”:



- Le decimos “Adiós” por el Chat:



- El bot nos responde “Adiós Sr. Saga Bayn”:



6.4. Bot en OsGrid

6.4.1. Código fuente de Bot en OsGrid

Vamos a ver el código fuente de un bot que se registra en la Grid de OsGrid.

- Código fuente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;
```

```
namespace bot2
{
class Program
   {
    public static GridClient Client = new GridClient();
     private static string first_name = "giaa";
     private static string last_name = "Quixote";
     private static string password = "*************";

    static void Main(string[] args)
    {
      Client.Network.OnConnected += new
       NetworkManager.ConnectedCallback(Network_OnConnected);

      Client.Settings.LOGIN_SERVER = "http://osgrid.org:8002/";

       if (Client.Network.Login(first_name, last_name, password, "My First
       Bot", "Your name"))
       {
        Client.Appearance.SetPreviousAppearance(true);
       }
       else
       {
         Console.WriteLine("I couldn't log in, here is why: " +
         Client.Network.LoginMessage);
         Console.WriteLine("press enter to close...");
         Console.ReadLine();
       }
     }
     static void Network_OnConnected(object sender)
     {
       Console.WriteLine("I'm connected to the simulator, going to greet everyone
       around me");
       Client.Self.Chat("Hello World!", 0, ChatType.Normal);
       Console.ReadLine();
       Console.WriteLine("Now I am going to logout of SL.. Goodbye!");
       Client.Network.Logout();
       Console.WriteLine("I am Loged out please press enter to close...");
       Console.ReadLine();
     }
   }
}
```

6.4.2. Explicación del código fuente

Vamos a explicar el código fuente.

➢ La siguiente línea de código vale para conectar el bot a OsGrid:

Client.Settings.LOGIN_SERVER = "http://osgrid.org:8002/";

6.4.3. Compilar y ejecutar la aplicación

➢ Vemos que nos conectamos a Osgrid:



Figura 183. Bot conectado a OsGrid

Se puede apreciar como nuestro avatar "saga Bayn" le dice "Hola" al bot giaa Quixote y el bot responde "Hola Sr. saga Bayn". Por lo tanto, concluimos que el bot funciona perfectamente en OsGrid.

6.5. Bot que envía un IM

6.5.1. Código fuente de bot que envía un IM

Vamos a ver el código fuente de un bot que se registra en la Grid SL y envía un IM.

- Código fuente del bot:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;
```

```
namespace Bot3
{
  class Program
  {
    public static GridClient Client = new GridClient();
    private static string first_name = “giaa”;
    private static string last_name = “Quixote”;
    private static string password = “***********”;

    static void Main(string[] args)
    {
      Client.Network.OnConnected += new
      NetworkManager.ConnectedCallback(Network_OnConnected);
      UUID target = new UUID(“ff974b5b-1f07-4946-9adf-7e9782a24958”);

      if (Client.Network.Login(first_name, last_name, password, “Program”,
      “e100048464@gmail.com”))
      {
        Client.Appearance.SetPreviousAppearance(true);
        Client.Self.InstantMessage(target, “Hola, sagaBayn”);
      }
      else
      {
        Console.WriteLine(“Yo no puedo conectarme aqui, es porque: “ +
        Client.Network.LoginMessage);
        Console.WriteLine(“Presione cualqueir tecla para cerrar...”);
        Console.ReadLine();
      }

    static void Network_OnConnected(object sender)
    {
      Console.WriteLine(“Yo estoy conectado al simulador, Yo voy a saludar a todo el
      mundo alrededor mio”);
      Client.Self.Chat(“Hola a todo el mundo!”, 0, ChatType.Normal);
      Console.ReadLine();
      Console.WriteLine(“Ahora yo voy a desconectarme de SL.. Adios!”);
      Client.Network.Logout();
      Console.WriteLine(“Yo estoy desconectado, por favor presione enter
      para cerrar...”);
      Console.ReadLine();
    }
  }
}
```

6.5.2. Explicación del código fuente

Vamos a explicar el código fuente.

- La siguiente línea de código vale guardar en una variable de tipo UUID el número identificativo del avatar "saga Bayn":

  UUID target = new UUID("ff974b5b-1f07-4946-9adf-7e9782a24958");

  En Second Life, todos los objetos tienen un número identificativo.

- Con la siguiente línea mandamos un IM al avatar Saga Bayn (pues target contiene la UUID de saga Bayn) con el texto "Hola, saga Bayn":

  Client.Self.InstantMessage(target, "Hola, sagaBayn");

6.5.3. Ejecutar la aplicación

Vamos a ejecutar la aplicación.

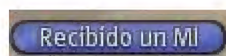
- Entramos en Second Life:



- Conectamos el bot:



➤ Vemos que nos ha enviado un IM:



6.6. Bot que vuela, se sienta y se levanta

6.6.1. Código fuente de bot que vuela, se sienta y se levanta

Vamos a ver el código fuente de un robot que se registra en la Grid SL y que:

- Cuando le decimos "volar" el bot se pone a volar.
- Cuando le decimos "dejar de volar" el bot aterriza.
- Cuando le decimos "siéntate" el bot se sienta en el suelo.
- Cuando le decimos "levántate" el bot se levanta del suelo.

- Código fuente del bot:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;

namespace Bot3
{
  class Program
   {
    public static GridClient Client = new GridClient();
     private static string first_name = "giaa";
```

```
private static string last_name = “Quixote”;
private static string password = “***********”;

static void Main(string[] args)
{
 Client.Network.OnConnected += new
 NetworkManager.ConnectedCallback(Network_OnConnected);
 if (Client.Network.Login(first_name, last_name, password, “Program”,
  “e100048464@gmail.com”))
  {
   Client.Appearance.SetPreviousAppearance(true);
    ClientSelf.OnChat += new AgentManager.ChatCallback(Self_OnChat);
  }

  else
  {
   Console.WriteLine(“Yo no puedo conectarme aqui, es porque: “ +
   Client.Network.LoginMessage);
   Console.WriteLine(“Presione cualqueir tecla para cerrar...”);
   Console.ReadLine();
  }
}

 static void Network_OnConnected(object sender)
  {
    Console.WriteLine(“Yo estoy conectado al simulador, Yo voy a saludar a
    todo el mundo alrededor mio”);
    Client.Self.Chat(“Hola a todo el mundo!”, 0, ChatType.Normal);
    Console.ReadLine();
    Console.WriteLine(“Ahora yo voy a desconectarme de SL.. Adios!”);
    Client.Network.Logout();
     Console.WriteLine(“Yo estoy desconectado, por favor presione enter
     para cerrar...”);
     Console.ReadLine();
   }

  static void Self_OnChat(string message, ChatAudibleLevel audible, ChatType
  type, ChatSourceType sourceType, string fromName, UUID id, UUID ownerid,
  Vector3 position)
  {
    string mensaje_minuscula = message.ToLower();

    if (mensaje_minuscula.Equals(“volar”))
    {
       Client.Self.Fly(true);
    }
    else if (mensaje_minuscula.Equals(“dejar de volar”))
    {
```

```
            Client.Self.Fly(false);
        }
        else if (message.Equals("levantate"))
        {
            Client.Self.Stand();
        }
        else if (message.Equals("sientate"))
        {
            Client.Self.SitOnGround();
        }
      }
    }
}
```

6.6.2. Explicación del código fuente

Vamos a explicar el código fuente.

- La siguiente línea de código vale para que el bot se ponga a volar:

```
Client.Self.Fly(true);
```

- La siguiente línea de código vale para que el bot aterrice:

```
Client.Self.Fly(false);
```

- La siguiente línea de código vale para que el bot se siente:

```
Client.Self.SitOnGround();
```

- La siguiente línea de código vale para que el bot se levante si esta sentado:

```
Client.Self.Stand();
```

6.6.3. Ejecutar la aplicación

Vamos a ejecutar la aplicación.

- Vemos como el avatar está con nosotros:



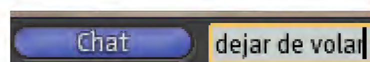
- Introducimos en el chat la palabra "volar":



- Y vemos que el avatar vuela:



- Introducimos en el chat "dejar de volar":



- Y vemos que nuestro bot deja de volar:



- Ahora le ordenamos que se siente en la tierra:



- Vemos que el avatar se sienta en la tierra:



- Ahora le decimos que se levante:



- Y vemos como el avatar se levanta:



6.7. Bot que me sigue

6.7.1. Código fuente de bot que me sigue

Vamos a ver el código fuente de un bot que se registra en la Grid SL y:

- Si se le dice "sígueme" el bot me sigue andando o volando.
- Si se le dice "no me sigas" el bot no sigue al avatar.
- Código fuente del bot:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;

namespace Bot3
{
 class Program
   {
     public static GridClient Client = new GridClient();
     private static string first_name = "giaa";
     private static string last_name = "Quixote";
```

```
private static string password = "***********";

private static float followDistance = 0;
public static bool followon = false;
public static string followName = "saga Bayn";

static void Main(string[] args)
{
 Client.Network.OnConnected += new
 NetworkManager.ConnectedCallback(Network_OnConnected);
 if (Client.Network.Login(first_name, last_name, password, "Program",
 "e100048464@gmail.com"))
  {
   Client.Appearance.SetPreviousAppearance(true);
   Client.Objects.OnObjectUpdated += new
   ObjectManager.ObjectUpdatedCallback(Objects_OnObjectUpdated);
   Client.Self.OnChat += new AgentManager.ChatCallback(Self_OnChatz
  }
  else
  {
   Console.WriteLine("Yo no puedo conectarme aqui, es porque: " +
   Client.Network.LoginMessage);
   Console.WriteLine("Presione cualqueir tecla para cerrar...");
   Console.ReadLine();
  }
}

static void Network_OnConnected(object sender)
{
 Console.WriteLine("Yo estoy conectado al simulador, Yo voy a saludar a todo el
 mundo alrededor mio");
 Client.Self.Chat("Hola a todo el mundo!", 0, ChatType.Normal);
 Console.ReadLine();
 Console.WriteLine("Ahora yo voy a desconectarme de SL.. Adios!");
 Client.Network.Logout();
 Console.WriteLine("Yo estoy desconectado, por favor presione enter
 para cerrar...");
 Console.ReadLine();
}

static void Objects_OnObjectUpdated(Simulator simulator, ObjectUpdate update,
ulong regionHandle, ushort timeDilation)
{
 if (followon == true)
 {
   if (!update.Avatar) { return; }
    Avatar av;
    Client.Network.CurrentSim.ObjectsAvatars.TryGetValue
```

```
(update.LocalID, out av);
  if (av == null) return;
  if (av.Name == followName)
   {
    Vector3 pos = av.Position;
    if (Vector3.Distance(pos, Client.Self.SimPosition) > followDistance)
     {
      int followRegionX = (int)(regionHandle >> 32);
      int followRegionY = (int)(regionHandle & 0xFFFFFFFF);
      int followRegionZ = (int)(regionHandle);
      ulong x = (ulong)(pos.X + followRegionX); ulong y = (ulong)(pos.Y +
      followRegionY);


     if (pos.Z > 1)
    {
     Client.Self.AutoPilotLocal(Convert.ToInt32(pos.X),
     Convert.ToInt32(pos.Y), pos.Z);
    }
    else
    {
      Client.Self.AutoPilotCancel();
    }
   }
  }
 }
}

 static void Self_OnChat(string message, ChatAudibleLevel audible,
 ChatType type, ChatSourceType sourceType, string fromName, UUID id,
 UUID ownerid, Vector3 position)
 {
   string mensaje_minuscula = message.ToLower();

   if (message.Equals("sigueme"))
    {
     followon = true;
     //Client.Self.Movement.AlwaysRun = true;
    }
   else if (message.Equals("no me sigas"))
    {
     followon = false;
    }
  }
 }
}
```

6.7.2. Explicación del código fuente

Vamos a explicar el código fuente.

- La siguiente línea de código sirve para indicar la distancia a la que el bot tiene que seguir a un avatar:

```
private static float followDistance = 0;
```

- La siguiente línea de código sirve para activar o desactivar que el bot siga a un avatar:

```
public static bool followon = false;
```

- La siguiente línea de código vale para indicar el avatar que va a seguir el bot:

```
public static string followName = "saga Bayn";
```

- El código fuente que hace que el bot me siga:

```
static void Objects_OnObjectUpdated(Simulator simulator, ObjectUpdate update,
ulong regionHandle, ushort timeDilation)
{
 if (followon == true)
 {
   if (!update.Avatar) { return; }
    Avatar av;
    Client.Network.CurrentSim.ObjectsAvatars.TryGetValue
    (update.LocalID, out av);
      if (av == null) return;
      if (av.Name == followName)
       {
        Vector3 pos = av.Position;


        if (Vector3.Distance(pos, Client.Self.SimPosition) > followDistance)
         {
          int followRegionX = (int)(regionHandle >> 32);
          int followRegionY = (int)(regionHandle & 0xFFFFFFFF);
          int followRegionZ = (int)(regionHandle);
          ulong x = (ulong)(pos.X + followRegionX); ulong y = (ulong)(pos.Y +
          followRegionY);

         if (pos.Z > 1)
        {
         Client.Self.AutoPilotLocal(Convert.ToInt32(pos.X),
         Convert.ToInt32(pos.Y), pos.Z);
        }
         else
        {
```

```
                    Client.Self.AutoPilotCancel();
                }
            }
        }
    }
}
```

- Código fuente que hace el bot me siga o deje de seguirme cuando se lo diga:

```
if (message.Equals("sigueme"))
{
  followon = true;
}
else if (message.Equals("no me sigas"))
{
  followon = false;
}
```

- La siguiente línea de código hace que el bot corra:

```
Client.Self.Movement.AlwaysRun = true;
```

- La siguiente línea de código hace que el bot deje de correr:

```
Client.Self.Movement.AlwaysRun = false;
```

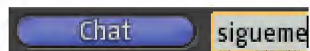
6.7.3. Ejecutar la aplicación

Vamos a ejecutar la aplicación.

- Vemos el bot:



- Ahora escribimos en el chat la palabra "sígueme":



- Vemos como el bot se ha puesto a nuestro lado:



- Damos al botón volar:



- Vemos que el bot nos sigue aunque estemos volando:



- Escribimos en el chat "volar":



- Vemos que el bot vuela:



- Vemos que el bot nos sigue volando:



6.8. Conexión de Bot y MySql

6.8.1. Crear una base de datos

Vamos a crear una base de datos para que el bot pueda coger la información de ella.

- Hacemos clic en el Acceso directo de WampServer, para conectarlo:



IMPORTANTE:

Hay que tener cuidado de no tener activo ningún otro programa que utilice también mysql, pues va a interferir en el funcionamiento de WampServer

Ej.: El programa xampp, que es un servidor php con myslql, que se utiliza en el programa Moodle, si lo tenemos instalado de forma local)

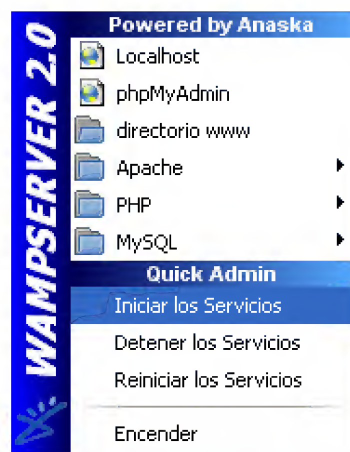
- Nos aparece un icono de WampServer:



- Hacemos clic en el icono de WampServer, nos sale un menú en el que seleccionamos **Encender**:



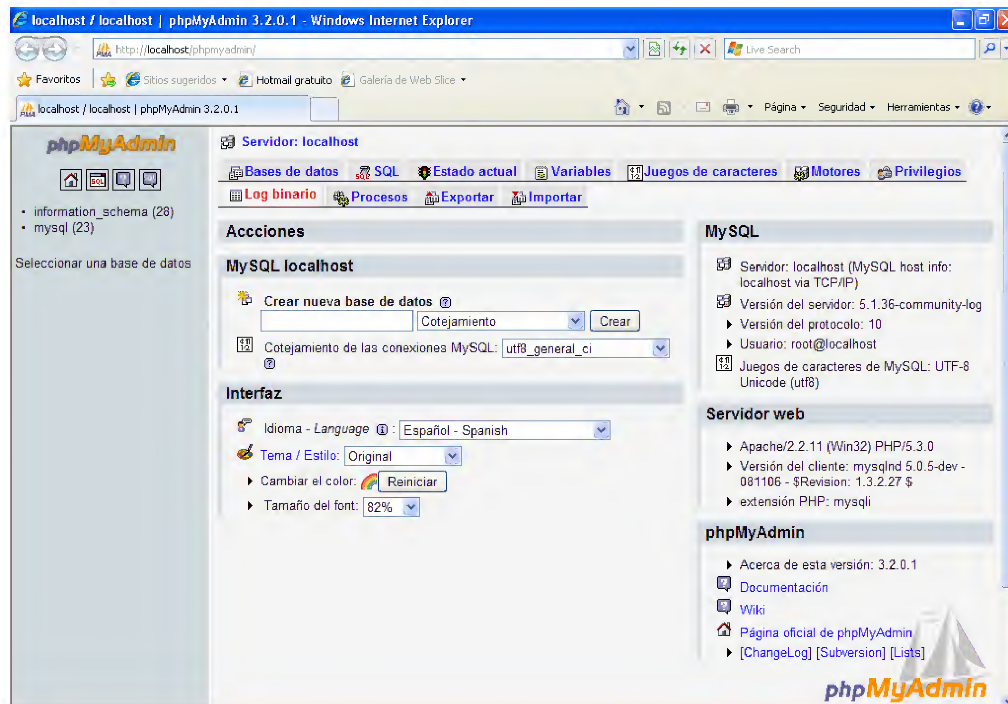
- Seleccionamos del menú de WampServer **Iniciar los Servicios**:



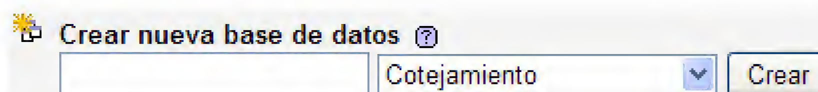
- Ya hemos Encendido el servidor, con la base de datos MySql:



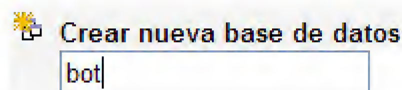
- Nos abre la siguiente ventana de administración de la base de datos mysql:



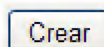
- Vamos a crear una nueva base de datos, para ello vamos a la zona de **Crear nueva base de datos**:



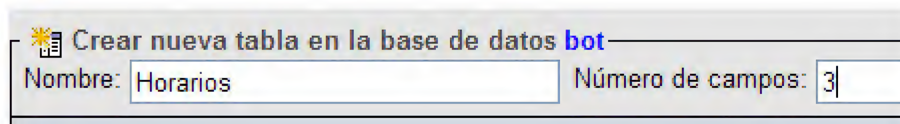
- Introducimos como nombre de la base de datos **bot**:



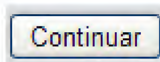
- Hacemos clic en el botón **Crear**:



- Creamos una tabla **Asignaturas** con 3 campos:



- Hacemos clic en el botón continuar:



- Introducimos la información de los dos campos que hemos creado, que son Asignatura, Horario, día_semana:

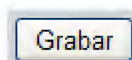
Campo	Asignatura	Horario	Dia_semana
Tipo	VARCHAR	VARCHAR	VARCHAR
Longitud/Valores*[1]	30	30	10
Predeterminado[2]	None	None	None
Cotejamiento			
Atributos			
Nulo			
Índice	PRIMARY	PRIMARY	PRIMARY
AUTO_INCREMENT			
Comentarios			

Figura 184. Claves primarias de la base de datos de la aplicación

Tanto "Asignatura" como "Horario" serán claves primarias, pues una asignatura puede tener varios horarios, pero para una asignatura no se puede repetir el mismo horario.

"Asignatura", "Horario" y "Día_semana" son claves, porque una asignatura, en el mismo horario se puede dar en distintos días de la semana.

- Hacemos clic en el botón **Grabar**:

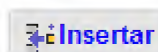


- Comprobamos que ya ha creado los campos en la tabla **bot**:

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
	Asignatura	varchar(30)	latin1_swedish_ci		No	*None*		
	Horario	varchar(30)	latin1_swedish_ci		No	*None*		
	Dia_semana	varchar(10)	latin1_swedish_ci		No	*None*		

Marcar todos/as / Desmarcar todos *Para los elementos que están marcados:*

- Hacemos clic en "Insertar", para introducir valores en cada uno de los campos:

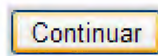


- Insertamos los primeros valores en los campos:

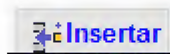
Campo	Tipo	Función	Nulo	Valor
Asignatura	varchar(30)			Procesadores de Lenguaje
Horario	varchar(30)			12:00
Dia_semana	varchar(10)			Lunes

Continuar

- Hacemos clic en el botón **Continuar**:



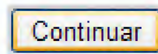
- Hacemos clic en el botón **Insertar**:



- Insertamos los segundos valores en los campos:

Campo	Tipo	Función	Nulo	Valor
Asignatura	varchar(30)			Procesadores de Lenguaje
Horario	varchar(30)			12:00
Dia_semana	varchar(10)			Miercoles

- Hacemos clic en el botón **Continuar**:



- Ya tenemos creada la tabla horarios con sus campos:

	Tabla	Acción	Registros [1]	Tipo	Cotejamiento	Tamaño	Residuo a depurar
	horarios		2	MyISAM	latin1_swedish_ci	2.1 KB	-
	1 tabla(s)	Número de filas	2	MyISAM	latin1_swedish_ci	2.1 KB	0 Bytes
	Marcar todos/as / Desmarcar todos		Para los elementos que están marcados:				

6.8.2. Establecimiento de una conexión

Una vez que el conector se ha instalado, puede utilizarlo para crear, modificar y eliminar conexiones a bases de datos MySQL. Para crear una conexión con una base de datos MySQL, realice los siguientes pasos.

- Hacemos clic en el Acceso directo de WampServer, para conectarlo:



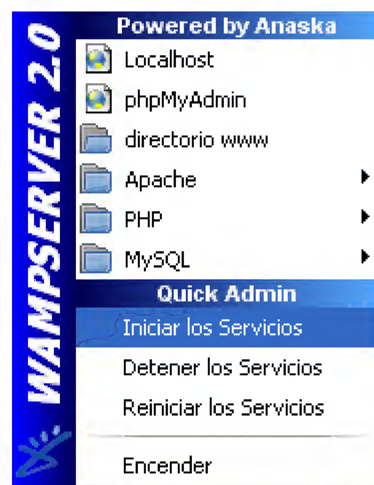
- Nos aparece un icono:



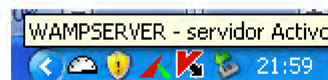
- Hacemos clic en el icono de WampServer, nos sale un menú en el que seleccionamos "Encender":



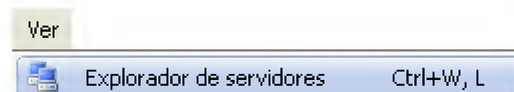
- Seleccionamos del menú de WampServer “Iniciar los Servicios”:



- Ya hemos Encendido el servidor, con la base de datos MySql:

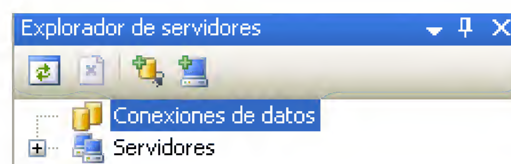


- Inicie Visual Studio y abra la ventana del Explorador de servidores:



(**Ver**, la opción **Explorador de servidores** en el menú principal de Visual Studio, o Ctrl + W, L teclas).

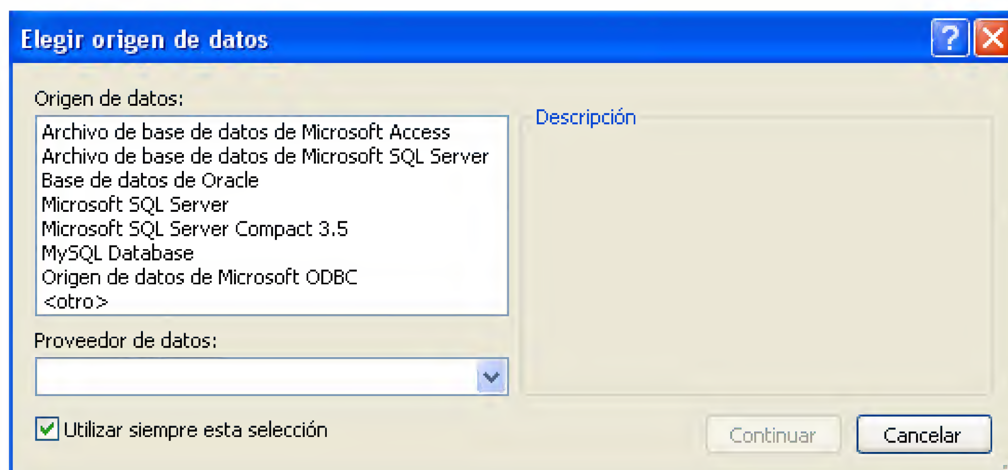
- Nos aparece la siguiente ventana:



- Haga clic derecho en el nodo Conexiones de datos, y elegimos la opción **Agregar conexión…**:



- Se abre el diálogo **Elegir origen de datos**:



- Elija la **MySQL Database** en la lista de fuentes de datos (o la <otra> opción, si está ausente de base de datos MySQL):

MySQL Database

- Elija **. NET Framework Data Provider for MySQL** en el cuadro combinado de los proveedores de datos:



- Hacemos clic en el botón continuar:

Continuar

- Entrada de Configuración (ajustes) de conexión:



- Introducimos el **Nombre de host del servidor**, que será localhost:

Server name: localhost

Figura 185. Nombre del Servidor mySQL

Utilizaremos localhost si el servidor MySQL se instala en la máquina local.

- Introduciremos el **Nombre de usuario**, que será el usuario **“root”**:

User name: root

- Introducimos el **Nombre de la base de datos**, que será “bot”**:**

Database name: bot

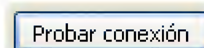
Figura 186. Nombre de la base de datos

Tenga en cuenta que debe especificar el nombre de esquema por defecto para abrir la conexión.

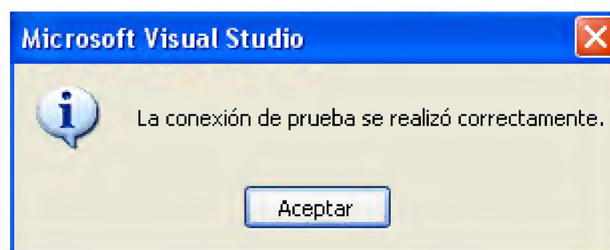
- Vemos el resultado:



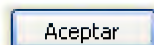
- Hacemos clic en el botón **Probar conexión**:



- Se abre una ventana emergente:

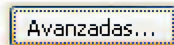


- Hacemos clic en el botón **Aceptar**:

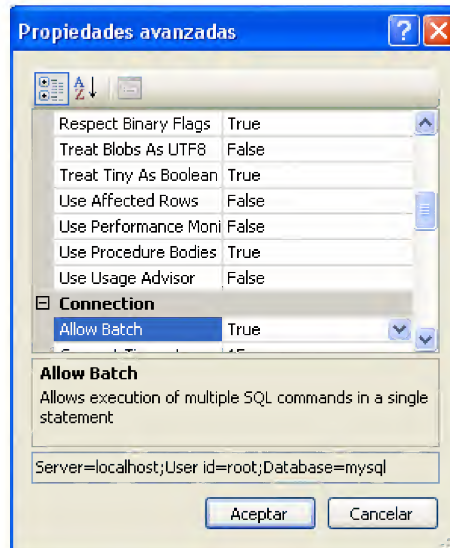


- También puede configurar el puerto para conectar con el servidor MySQL pulsando en el botón Opciones **avanzadas**. Para probar la conexión con el servidor MySQL, configure el nombre de host del servidor, el nombre de usuario y la contraseña y pulse el botón **Probar conexión**. Si la prueba tiene éxito, se abre el cuadro de diálogo de confirmación de éxito.
-

- Hacemos clic en el botón **Avanzadas**:

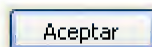


- Se nos abre la siguiente ventana:

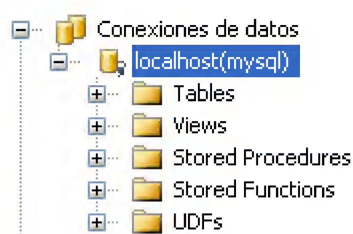


- Después de configurar todos los ajustes y probar la conexión, pulse **Aceptar**. La recién creada conexión se muestra en el Explorador de servidores. Ahora usted puede trabajar con el servidor MySQL a través de Explorador de servidores estándar de la GUI.

- Hacemos clic en el botón "**Aceptar** de **Agregar conexión**:



- Nos crea los siguientes datos de conexión:

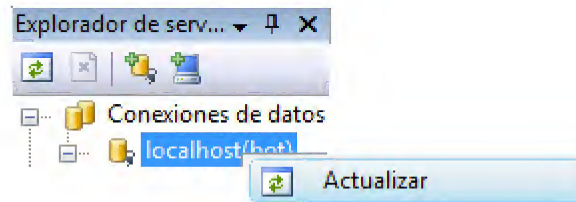


- Ya tenemos nuestra "Conexión de datos" a "localhost(bot)":



- Después de que la conexión se ha establecido, todos los ajustes se guardan para su uso futuro.

- Al iniciar Visual Studio para la próxima vez, simplemente abra el nodo de conexión en el Explorador de servidores para establecer una conexión con el servidor MySQL de nuevo.



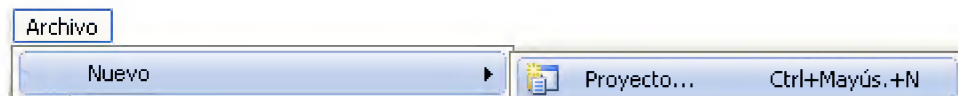
- Para modificar y eliminar una conexión, utilice el menú contextual del Explorador de servidores para el nodo correspondiente. Usted puede modificar cualquiera de los ajustes sólo sobrescribiendo los valores existentes por otros nuevos. Tenga en cuenta que la conexión podrá ser modificada o suprimida solo si no se activa el editor para parar sus objetos: de lo contrario pueden perder sus datos.

6.8.3. Conexión a base de datos

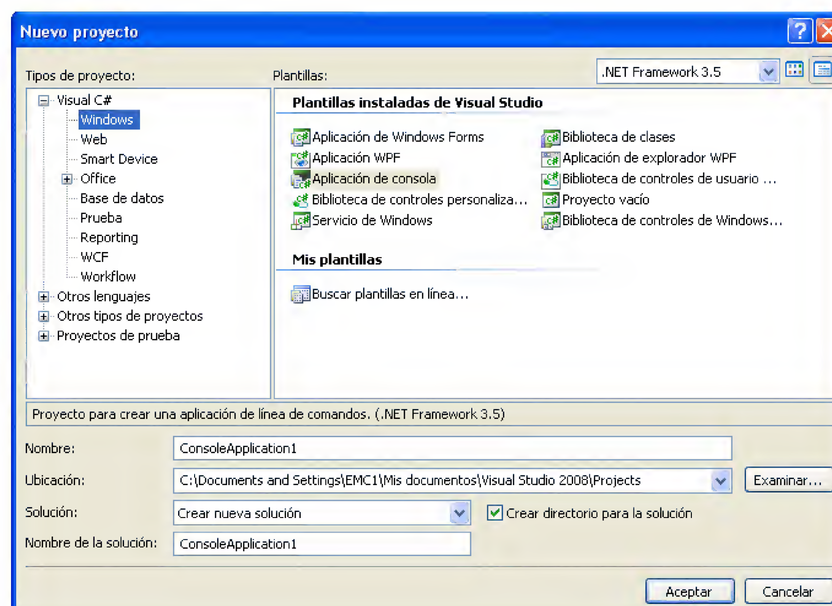
6.8.3.1. Creación de un proyecto

Vamos a crear un nuevo proyecto.

- Vamos a **Archivo → Nuevo → Proyecto**:



- Nos muestra la siguiente pantalla:



- Seleccionamos **Visual C# → Windows**:

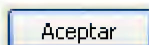


- Seleccionamos Aplicación de consola:



- Introducimos en el **Nombre** Bot3:

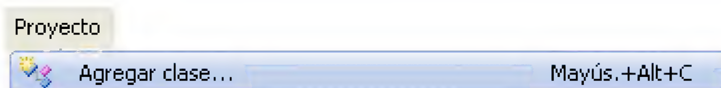
- Hacemos clic en el botón **Aceptar**:



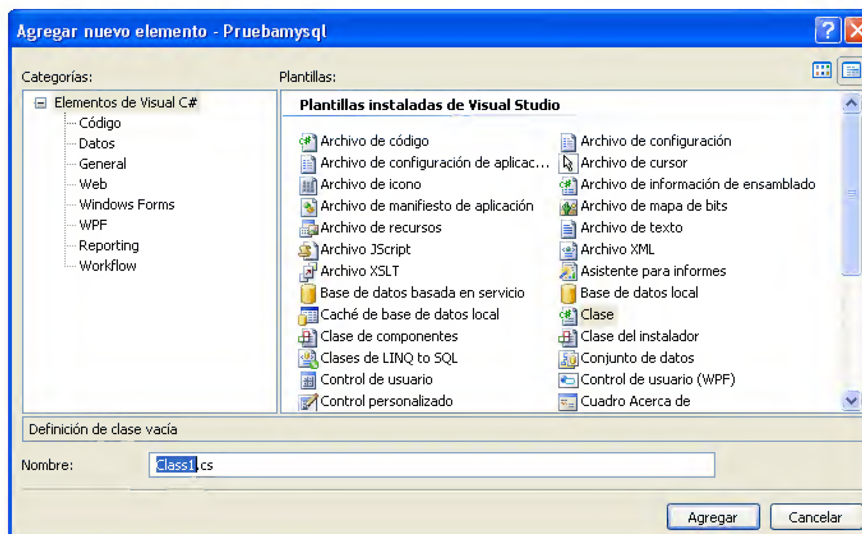
6.8.3.2. Crear la clase de conexión de MySql

Vamos a crear la clase Conectar_MySql.

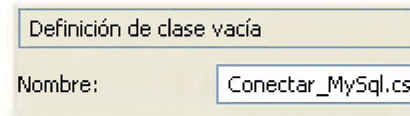
- En el menú **Proyecto**, haga clic en **Agregar clase**:



- Nos abre una nueva ventana:



- En el cuadro de diálogo **Agregar nuevo elemento**, en **Nombre de clase** escriba **Conectar_MySql** y, a continuación, haga clic en **Agregar**:



- Hacemos clic en el botón agregar:



- Nos crea la siguiente clase:



- Introducimos el siguiente código, en la ventana Vista Código:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.Types;
using MySql.Data.MySqlClient;

namespace Conexion_bot
{
class Conectar_MySql
  {
   private string connStr;
   private MySqlConnection conn;

   public Conectar_MySql()
    {
      //conectar con Mysql
      //se conecta con MySQl
      //Server: nombre del servidor
      //Database:nombre de la base de datos (no va con ningun tipo de
      extencion)
      //Uid:Usuario(Standard=root) no soporta el usuario vacío
      //Pwd:clave de acceso (si la tiene)
      //Port:default=3306(si no se coloca se toma el default)
      connStr = "Server=localhost;Database=bot;Uid=root;Pwd= ;Port=3306";
      conn = new MySqlConnection(connStr);
    }

public MySqlConnection Cnx
 {
   get
     {
```

```
                conn.Open();
                return conn;
            }
        }
    }
}
```

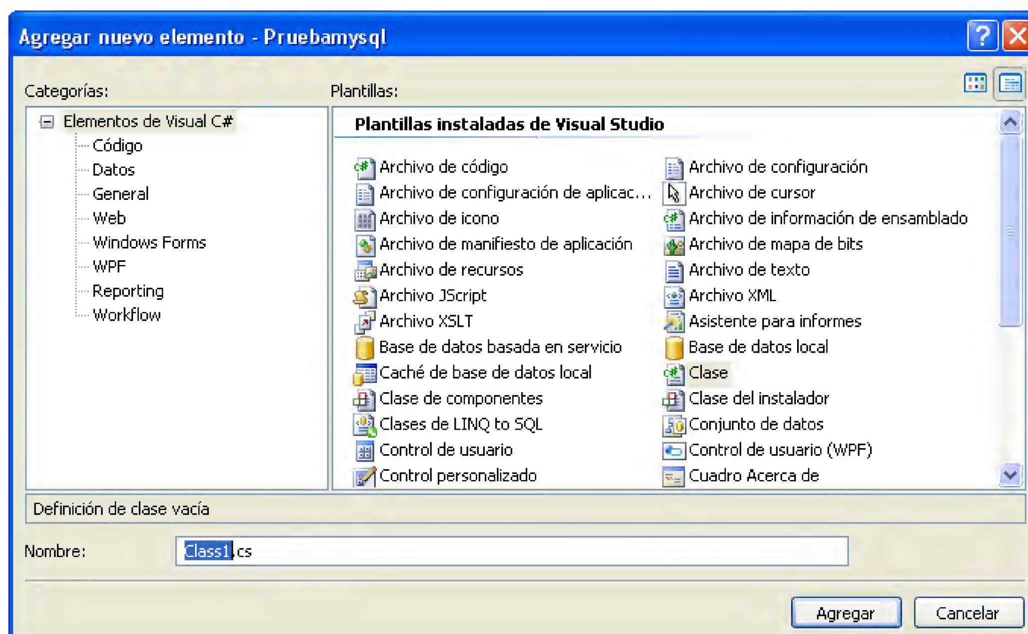
6.8.3.3. Creamos la clase de ejecución de MySql

Ahora vamos a crear la clase "Ejecutar SQL".

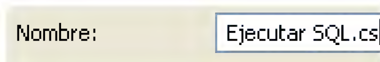
- En el menú **Proyecto**, haga clic en **Agregar clase**:



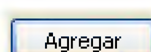
- Nos abre una nueva ventana:



- En el cuadro de diálogo **Agregar nuevo elemento**, en **Nombre de clase** escriba **Conectar_MySql** y, a continuación, haga clic en **Agregar**:



- Hacemos clic en el botón agregar:



- Nos crea la siguiente clase:



* Introducimos el siguiente código, en la ventana Vista Código:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.Types;
using MySql.Data.MySqlClient;

namespace Conexion_bot
{
 class Ejecutar_SQL
  {
    private static MySqlCommand command;
    private static Conectar_MySql Coneccion;

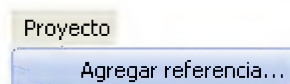
    //Creo un metodo que recibe la sentencia SQL y retorna un objeto de
    //MySqlDataReader

   public static MySqlDataReader EjecutarSQL(string Sentencia_SQL)
    {
    //INSTANCIA DEL CONECTOR
    Coneccion = new Conectar_MySql();
     command = new MySqlCommand(Sentencia_SQL, Coneccion.Cnx);
     return command.ExecuteReader();
    }
  }
}
```

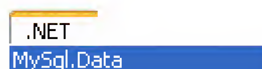
6.8.3.4. Añadimos las librerías

Vamos a añadir las librerías al proyecto.

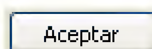
* Vamos a **Proyecto → Agregar referencia**:



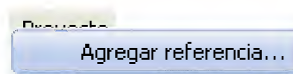
* Nos abre una nueva ventana en la que seleccionamos la ficha **.NET**, y seleccionamos **MySql.Data**:



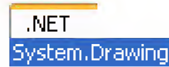
* Hacemos clic en el botón **Aceptar**:



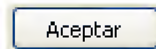
* Vamos a **Proyecto → Agregar referencia**:



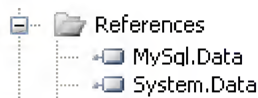
* Nos abre una nueva ventana en la que seleccionamos la ficha **.NET**, y seleccionamos **MySql.Data**:



* Hacemos clic en el botón **Aceptar**:



* Vemos que se han agregado las referencias:



6.9. Bot que responde a preguntas mediante información de la base de datos

6.9.1. Código fuente

Vamos a ver el código fuente de un bot que se registra en la Grid SL y responde a preguntas mediante información de la base de datos.

* Código fuente del bot:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using MySql.Data.Types;
using MySql.Data.MySqlClient;
using System.ComponentModel;
using System.Drawing;


namespace Conexion_bot
{
  class Program
    {
      static void Main(string[] args)
        {
          string Tabla = “horarios”;
          string Asignatura = “”;
```

```
            string Horario = “”;
            string dia_semana = “”;

            MySqlDataReader Resul;
            /* Ejecutar_SQL devuelve de su metodo de tipo MysqlDataReader
             * con este puedo manipular los datos sin estar conectado a la bd
            */
            Resul = Ejecutar_SQL.EjecutarSQL(“SELECT asignatura, horario,
dia_semana FROM “ + Tabla);

            /* Resul.Read es un metodo boleano que devuelve true si no es
             * fin de archivo
             */

            while (Resul.Read())
            {
              /* es conveniente no colocar el numero de columna sino
               * el nombre debido que mas adelante se necesite agregar otra
               * columna no tengamos problemas con este el indice de columnas
              */

              Asignatura = Resul.GetString(“asignatura”);
              Horario = Resul.GetString(“horario”);
              dia_semana = Resul.GetString(“dia_semana”);

              Console.WriteLine(Asignatura + “ “ + Horario + “ “ + dia_semana);
              //coloque un msgbox solo para mantener el ejemplo sencillo
            }

            Console.ReadLine(); //Espera a que el usuario presione una tecla.
        }
    }
}
```

6.9.2. Explicación del código fuente

Vamos a explicar el código fuente.

- Las siguientes líneas de código son las librerías que necesitamos para utilizar la base de datos:

```
using MySql.Data.Types;
using MySql.Data.MySqlClient;
```

- La siguiente línea de código contiene los datos necesarios para identificar el **servidor** (Server), la **base de datos** (Database) y como que **usuario** (Uid) nos conectamos a la base de datos:

```
connStr = “Server=localhost;Database=bot;Uid=root;Pwd= ;Port=3306”;
```

- La siguientes líneas de código son variables que las utilizamos para realizar la consulta SQL:

```
string Tabla = “horarios”;
string Asignatura = “”;
string Horario = “”;
string dia_semana = “”;
```

- La siguiente línea de código es una variable que almacenará el valor de ejecutar una sentencia SQL:

```
MySqlDataReader Resul;
```

- La siguiente línea de código llama al método Ejecutar_SQL que retorna un objeto de tipo MySqlDataReader que contiene el resultado de ejecutar una sentencia SQL:

```
Resul = Ejecutar_SQL.EjecutarSQL(“SELECT asignatura, horario,
dia_semana FROM “ + Tabla);
```

- La siguiente línea de código vale para ir leyendo los distintos resultados de las ejecución de una sentencia SQL:

```
while (Resul.Read())
```

- La siguiente línea de código vale para coger del resultado de la ejecución de una sentencia SQL el atributo que queremos coger:

```
Asignatura = Resul.GetString(“asignatura”);
```

6.9.3. Ejecutar la aplicación

Vamos a ejecutar la aplicación.

- Estamos en Second Life como el avatar “saga Bayn” y vemos que el bot “giaa Quixote” nos saluda, mediante el Chat y nos dice: “Hello World!”:

giaa Quixote: Hola a todo el mundo!

- Nuestro avatar “saga Bayn” le dice “hola” al bot “giaa Quixote” mediante el chat:

Chat hola

- El bot nos responde “Hola Sr. Saga Bayn”:

saga Bayn: hola
giaa Quixote: Hola Sr. saga Bayn

- Nuestro avatar “saga Bayn” le dice “ádios” al bot “giaa Quixote” mediante el chat:



- El bot nos responde “ádios Sr. Saga Bayn”:

saga Bayn: adios
giaa Quixote: Adios Sr. saga Bayn

- Nuestro avatar “saga Bayn” le dice “horarios” al bot “giaa Quixote” mediante el chat:



- El bot nos responde:

saga Bayn: horarios
giaa Quixote: procesadores de lenguaje 12:00 lunes
giaa Quixote: procesadores de lenguaje 12:00 miercoles

6.10. Bot que habla

6.10.1. Configurar SAPI

Vamos a configurar el SAPI.

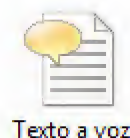
- Hacemos clic en el botón de inicio de Windows:



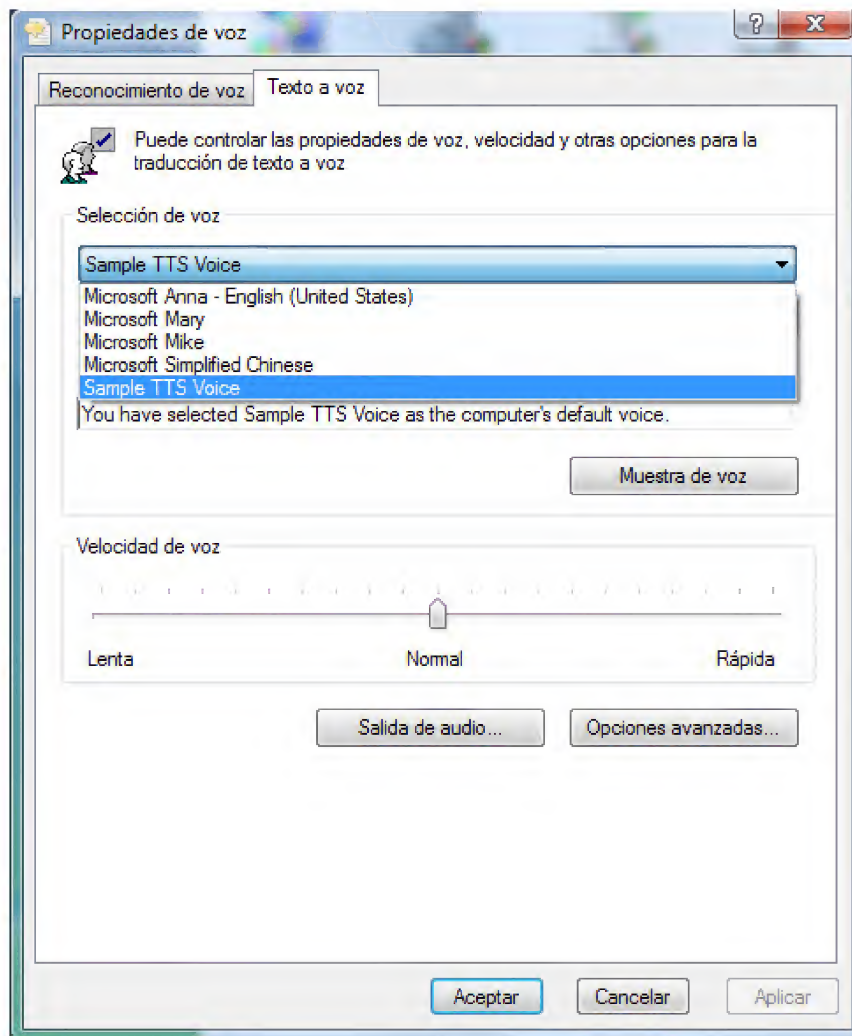
- Hacemos clic en el panel de control:



- Hacemos clic en el icono “Texto a voz”:



- Nos aparece la siguiente ventana, en la cual podemos apreciar que las voces disponibles están en ingles:



- No cualquier voz española vale, tiene que ser una voz española que valga para el programa "Texto a voz" para Microsoft.

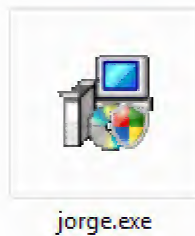
- Para conseguir voces en español que valgan para el programa "Texto a voz" para Microsoft vamos a la siguiente dirección Web:

http://www.loquendo.com/es/

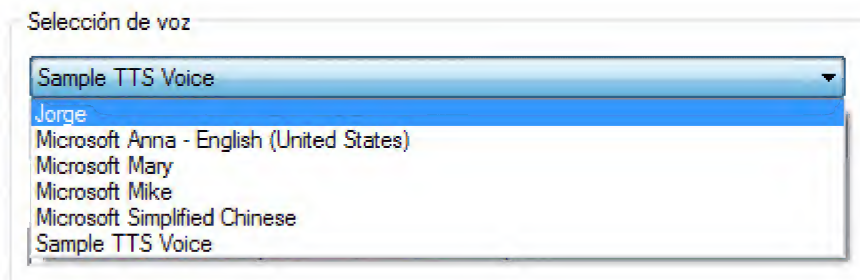
- Hemos comprado la voz de **Jorge**:



- Descomprimimos el fichero, y lo ejecutamos:



- Ahora comprobamos las voces disponibles en "Texto a voz" de Microsoft y comprobamos que esta la voz de Jorge:



- Como trabajamos en Microsoft Windows Vista, no me hace falta instalar el SAPI5 (programa para que hable el ordenador) pues ya está instalado.

- Si necesitamos instalar el SAPI5 vamos a la siguiente dirección Web:

http://www.nextup.com/sapi.html

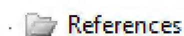
- Y nos aparece la siguiente información donde podremos descargarlo:

Downloading and installing SAPI5
Versions of windows 2000, and XP typically already have SAPI5 installed, but you may wish to install it again if you are having problems, or if you are using a different windows version. Download and run the file below. If you have problems contact us, as you may not have windows installer on your computer and we can provide help with that.
http://www.webaloud.com/files/Microsoft-English-TTS-51.msi

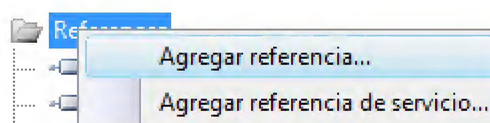
6.10.1.2. Añadimos las librerías para usar el SAPI

Ahora seleccionamos las librerías necesarias para utilizar el SAPI.

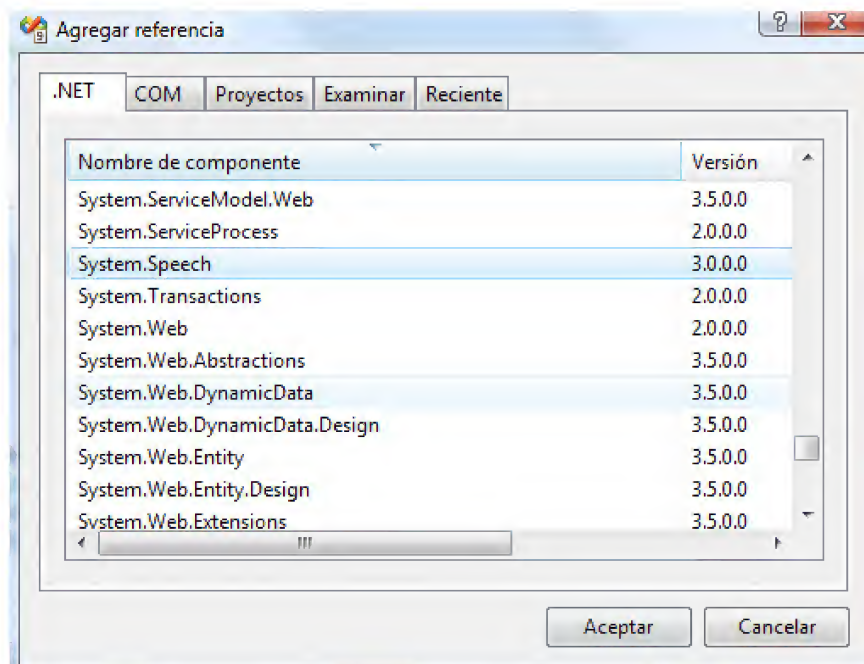
- Vamos a **References**:



- Pinchamos en referencias con el botón derecho del ratón, y nos aparece el siguiente menú, pinchamos en **Agregar referencia…**:



- Nos aparece el siguiente menú:



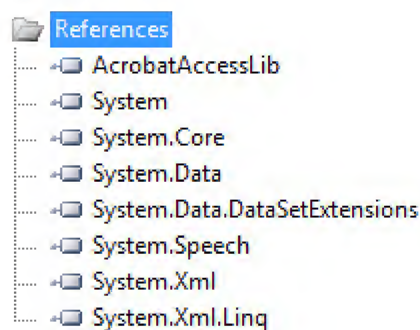
- Seleccionamos la opción **System.Speech**:

System.Speech

- Damos al botón **Aceptar**:

Aceptar

- Comprobamos que la librería **System.Speech** este añadida:



- Añadimos el namespace **System.Speech.Synthesis**:

```
using System.Speech.Synthesis;
```

6.10.1.3. Código fuente para hablar

Vamos a ver el código fuente para que el ordenador hable.

- Código fuente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Speech.Synthesis;

namespace ConsoleApplication1
{
 class Program
  {
   static void Main()
    {
     try
      {
       SpeechSynthesizer synth = new SpeechSynthesizer();
         synth.SelectVoice("Jorge");
         synth.Speak("Mi nombre es Eduardo Rojo Sánchez.");
      }
     catch
     {
      Console.Write("Error");
      Console.ReadLine(); //Espera a que el usuario presione una tecla.
      }
    }
   }
}
```

6.10.1.4. Código fuente para que el bot hable

Vamos a ver el código fuente de un bot que se registra en la Grid SL y dice hace que el bot hable.

- Código fuente del bot:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using OpenMetaverse;
using System.Speech.Synthesis;

  namespace Bot3
```

```
{
class Program
  {
    public static GridClient Client = new GridClient();
    private static string first_name = "giaa";
    private static string last_name = "Quixote";
    private static string password = "***********";

    static void Main(string[] args)
    {
      Client.Network.OnConnected += new
      NetworkManager.ConnectedCallback(Network_OnConnected);
       if (Client.Network.Login(first_name, last_name, password, "Program",
       "e100048464@gmail.com"))
       {
        Client.Appearance.SetPreviousAppearance(true);
       }
       else
       {
         Console.WriteLine("Yo no puedo conectarme aqui, es porque: " +
         Client.Network.LoginMessage);
         Console.WriteLine("Presione cualqueir tecla para cerrar...");
         Console.ReadLine();
       }
    }

   static void Network_OnConnected(object sender)
    {
      Console.WriteLine("Yo estoy conectado al simulador, Yo voy a saludar a
       todo el mundo alrededor mio");
       Client.Self.Chat("Hola a todo el mundo!", 0, ChatType.Normal);
      Console.ReadLine();
       Console.WriteLine("Ahora yo voy a desconectarme de SL.. Adios!");
       Client.Network.Logout();
       Console.WriteLine("Yo estoy desconectado, por favor presione enter
       para cerrar...");
       Console.ReadLine();
    }

static void Self_OnChat(string message, ChatAudibleLevel audible, ChatType type,
ChatSourceType sourceType, string fromName, UUID id, UUID ownerid, Vector3
position)
 {
  string mensaje_minuscula = message.ToLower();
  SpeechSynthesizer synth = new SpeechSynthesizer();
  synth.SelectVoice("Jorge");
    UUID typing = new UUID("c541c47f-e0c0-058b-ad1a-d6ae3a4584d9");
```

```
if (mensaje_minuscula.Equals("hola"))
 {
   synth.Speak("Hola Sr. " + fromName);
    Client.Self.AnimationStart(typing, true);
    Client.Self.Chat("Hola Sr. " + fromName, 0, ChatType.Normal);
    Client.Self.AnimationStop(typing, true);
  }
else if (mensaje_minuscula.Equals("adiós"))
  {
    synth.Speak("Adiós Sr. " + fromName);
    Client.Self.AnimationStart(typing, true);
    Client.Self.Chat("Adiós Sr. " + fromName, 0, ChatType.Normal);
    Client.Self.AnimationStop(typing, true);
  }
 }
}
}
```

6.10.1.5. Explicación del código fuente para que el bot hable

Vamos a explicar el código fuente para que el bot hable.

- Añadimos la siguiente librería:

```
using System.Speech.Synthesis;
```

- En el método Self_OnChat (método para el chat) añadimos las siguientes líneas de código para seleccionar la voz del ordenador que va a hablar:

```
SpeechSynthesizer synth = new SpeechSynthesizer();
synth.SelectVoice("Jorge");
```

- La siguiente línea de código vale para que cuando en el chat escribamos la palabra Hola, el bot responda mediante sonido Hola Sr más el nombre del avatar que nos ha dicho hola:

```
if (mensaje_minuscula.Equals("Hola"))
  {
    synth.Speak("Hola Sr. " + fromName);
   }
```

6.10.1.6. Código fuente para guardar sonido en WAV

Vamos a ver el código fuente para que el ordenador guarde el sonido de hablar en un fichero WAV.

- Código fuente:

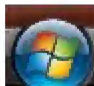
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Speech.Synthesis;
namespace ConsoleApplication1
{
class Program
  {
    static void Main()
     {
       try
         {
          SpeechSynthesizer synth = new SpeechSynthesizer();
           synth.SetOutputToWaveFile("c:\\Users\\100048464\\Desktop\\
           message.wav");
           synth.SelectVoice("Jorge");
           synth.Speak("Hola mi nombre es eduardo.");
         }
       catch
         {
          Console.Write("Error");
          Console.ReadLine(); //Espera a que el usuario presione una tecla.
         }
     }
   }
}
```

6.10.2. Reconocimiento de voz

6.10.2.1. Activar el reconocimiento de voz

Vamos a activar el reconocimiento de voz.

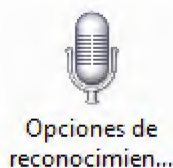
- Hacemos clic en el **botón de inicio** de Windows:



- Hacemos clic en **Panel de control**:



- Hacemos clic en el icono **Opciones de reconocimiento de voz**:



- Realizamos el tutorial de reconocimiento de voz:



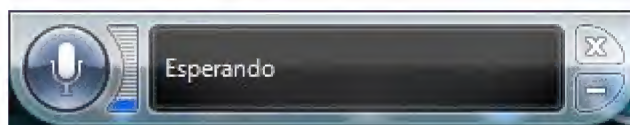
Realizar el tutorial del reconocimiento de voz
Aprender a usar el equipo con la voz. Aprender los comandos básicos y el dictado.

- Y después hacemos clic en **iniciar reconocimiento de voz**:



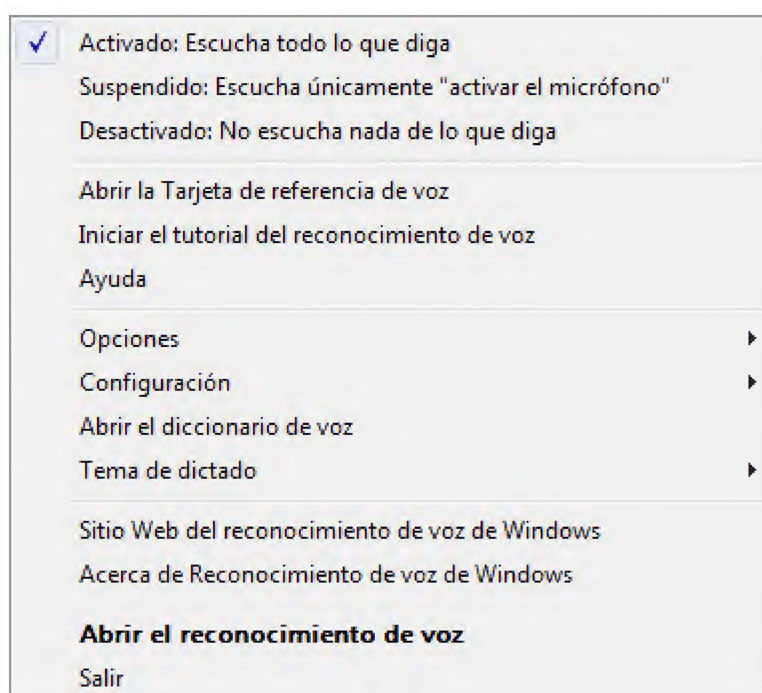
Iniciar reconocimiento de voz
Empezar a utilizar la voz para controlar el equipo.

- Se abre el siguiente panel:



- Decimos **Activar el micrófono** para que el micrófono se active.

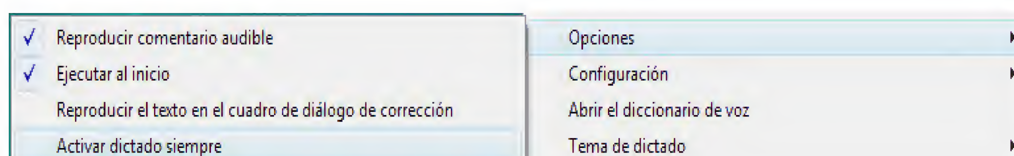
- Le decimos al reconocedor de voz la siguiente frase “mostrar opciones de voz“, y nos muestra el siguiente panel:



- Seleccionamos opciones:

Opciones

- Se nos abre un nuevo panel:



- Seleccionamos la opción **Activar dictado siempre**:

Activar dictado siempre

6.10.2.2. Activar el habla en Second Life

Vamos a activar el habla en Second Life.

- Abrimos el explorador de Second Life:



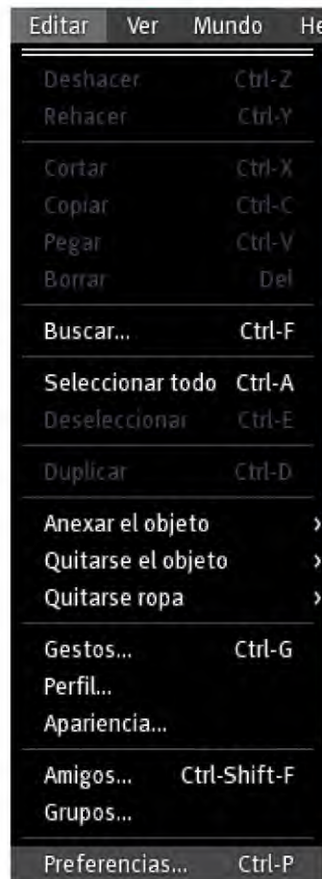
- Comprobamos el aspecto del avatar, y comprobamos que no tiene activado el **chat de voz**, pues no tiene un punto blanco encima de la cabeza del avatar:



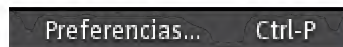
- Hacemos clic en editar:

Editar

- Se nos abre el siguiente menú:



Hacemos clic en **Preferencias**:



- Y nos aparece la siguiente pantalla:

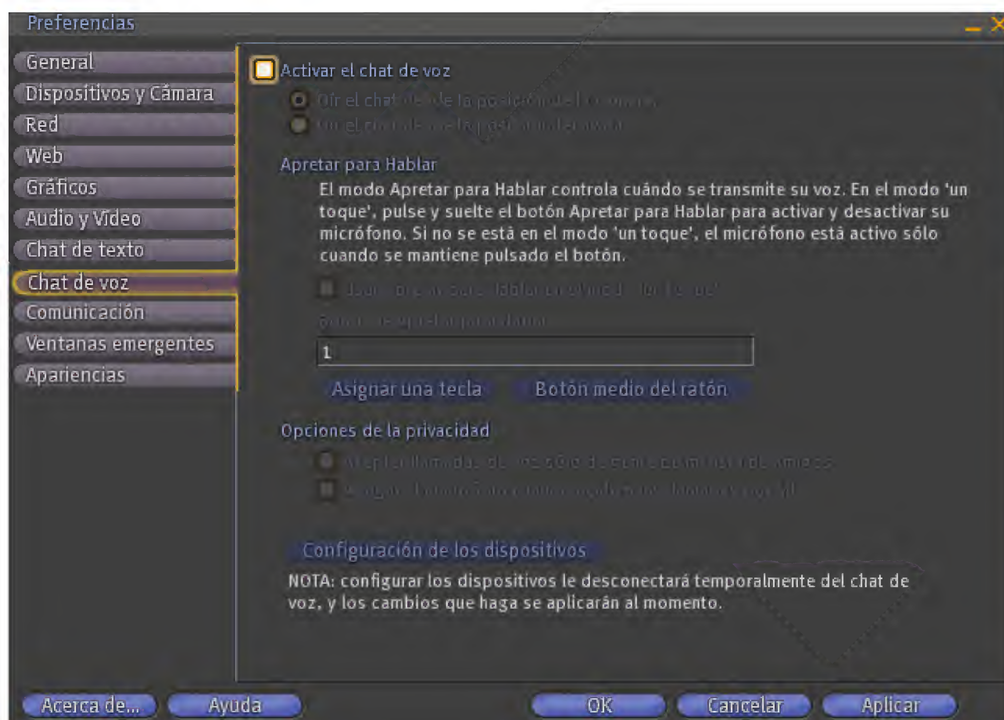


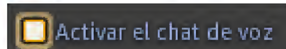
Figura 6.5. Chat de voz del visor de Second Life

En esta pantalla configuramos el chat de voz.

- Hacemos clic en la pestaña **Chat de voz**:



- Hacemos clic en la opción de **Activa el chat de voz**:



- Comprobamos que encima de la cabeza del avatar hay una bola blanca:



- Vemos el siguiente cuadro de para poder hablar:



- Hacemos clic en el botón **Hablar**:



  - Y vemos cuando hablamos desde el avatar salen ondas verdes:



6.10.2.3. Prueba de reconocimiento de voz y hablar

Vamos a hacer las pruebas de reconocimiento de voz y hablar.

- Entramos en Second Life:



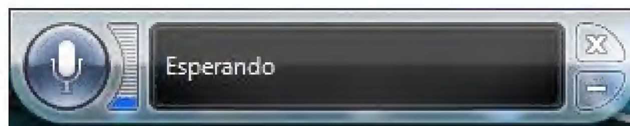
- Vemos el siguiente cuadro de para poder hablar:



- Hacemos clic en el botón **Hablar**:



- Activamos el reconocimiento de voz:



- Decimos la palabra **Hola**:

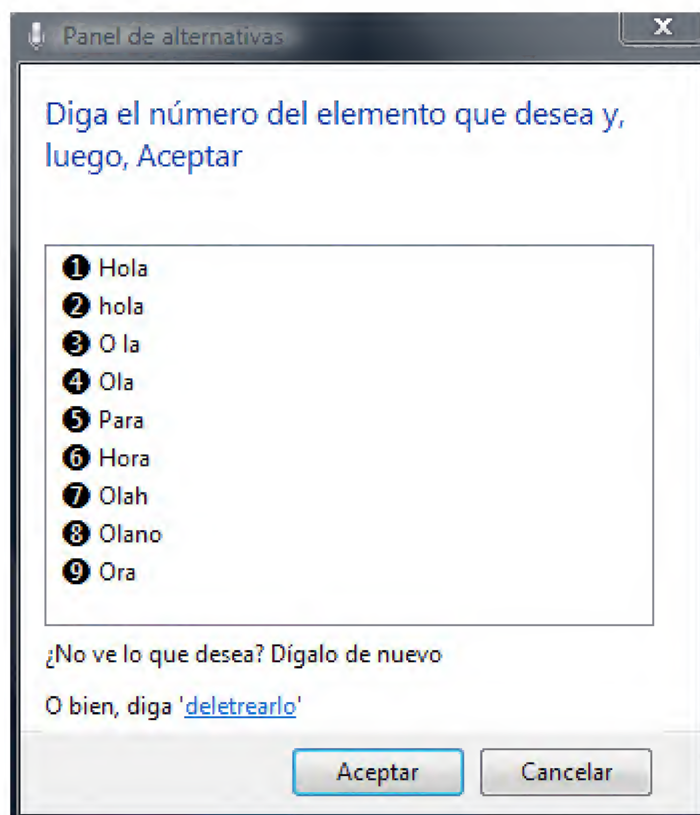
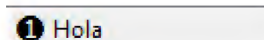


Figura 187. Listado de palabras que nos da el reconocimiento de voz

Nos sale el siguiente listado con la lista de palabras que podemos elegir.

- Decimos **uno** y se selecciona de la lista el primer elemento:



- Cuando hablamos el sistema de reconocimiento de voz, escribirá en el chat lo que nosotros hemos dicho:



- Decimos la palabra **Intro** para que se pulse la tecla Intro y de esta manera se envía la palabra **Hola** por el chat:



- De esta manera le hemos enviado la información al bot tanto por voz como por texto en el chat, y de esta manera se envía de la mejor forma posible.

- El sistema de reconocimiento de voz de Microsoft Windows Vista es muy bueno, y al ser propio del sistema operativo ningún usuario lo tiene que instalar en su ordenador.

- Algo muy importante es que al ser el sistema de reconocimiento de voz de Microsoft propio de una persona, al entrenarlo, ese sistema de reconocimiento de voz reconocerá la voz de esa persona de la mejor manera posible, pues está entrenado para esa persona, y de esta manera, pasara a texto lo que le diga con la mayor precisión que se puede.

- Si intentáramos reconocer la voz mediante el API habría dos problemas:

- **El sonido en Second Life.**

  El sonido dentro de Second Life es de muy baja calidad, e incluso para una persona es difícil comprender las palabras cuando otra persona la habla por tanto, si utilizamos un programa de ordenador para reconocer la voz mediante el sonido de baja calidad que nos proporciona Second Life, dará como resultado un mal reconocimiento de la voz, y como consecuencia el bot reconocería muy pocas palabras.

- **El entrenamiento del reconocedor de voz.**

  Como el reconocedor de voz del API no va a estar entrenado para la persona que se comunique con el bot, pues dará como resultado un reconocimiento de voz algo peor.

- Ahora vemos como el bot, al decirle hola, contesta mediante el chat y mediante la voz:

giaa Quixote: Hola Sr.saga Bayn

6.11. Bot que contesta a preguntas

6.11.1. Funcionamiento del bot

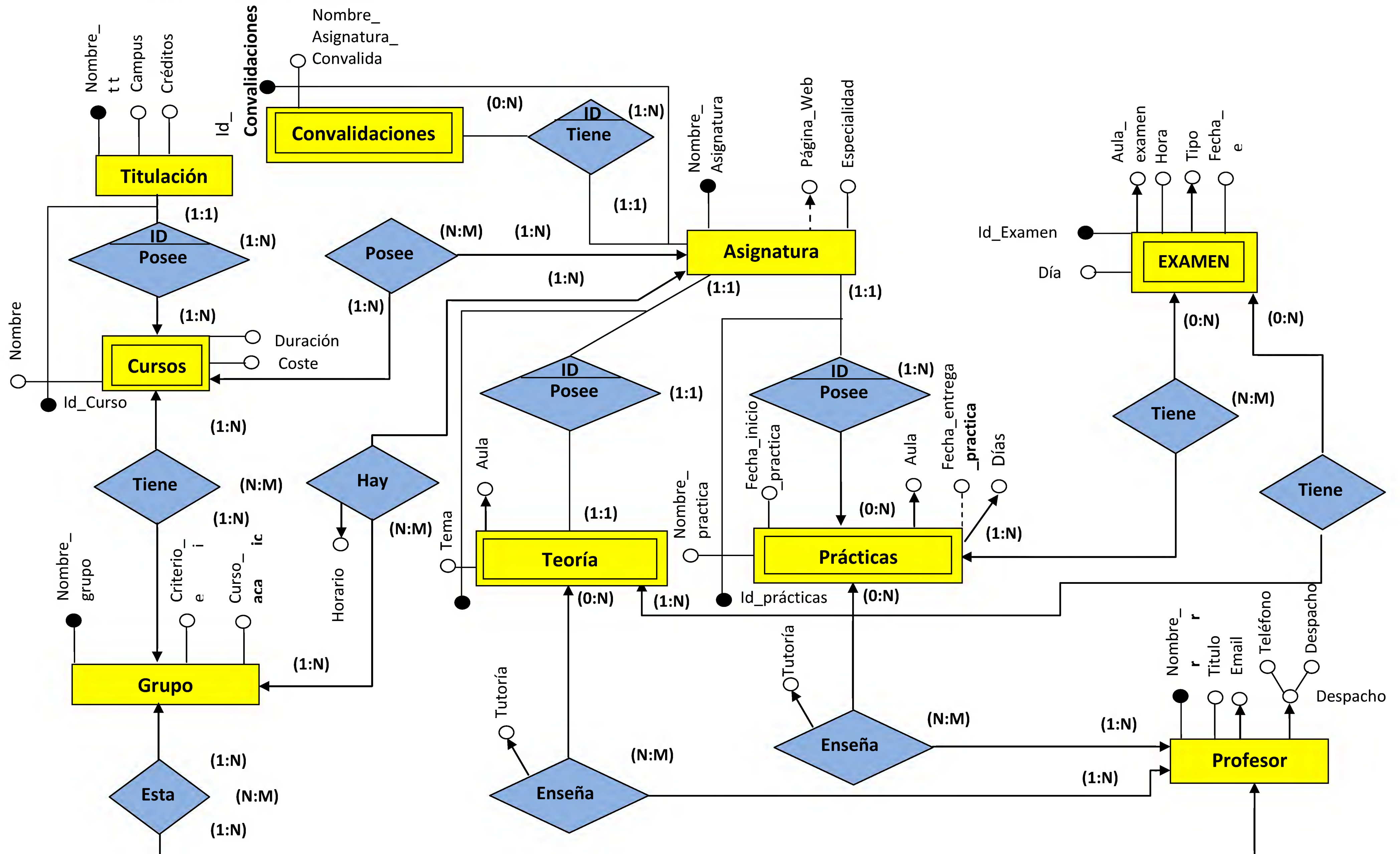
Vamos a describir el funcionamiento del bot, este bot tendrá todas las funcionalidades antes descritas anteriormente.

- Responde a preguntas.
- Se conecta a una base de datos.
- Tiene la capacidad de hablar.
- Se conecta a los mundos virtuales Second Life y OsGrid.
- Es capaz de envía IM.
- Vuela, se sienta y se levanta.
- Sigue a los avatares si se le ordena.

Y además en este bot hemos creado un **módulo de compresión del lenguaje** que es el programa que controla las repuestas del bot.

Este módulo extrae las palabras de la frase que ha recibido el avatar que son claves para determinar qué respuesta devolver y que tiene definidas en ficheros de categorías (tipos de consulta, asignaturas, profesores, titulaciones, etc.).

6.11.2. Modelo Entidad Relacional



- Hacemos una relación enseña entre profesor y teoría, porque así sabemos que tema de teoría enseña un profesor determinado.

- Hacemos una relación enseña entre profesor y práctica, porque así sabemos que practica enseña un profesor determinado.

  - Supuestos semánticos:

  - Tipo Examen: {test, escrito}

  - Un profesor tiene que dar por lo mínimo una asignatura en alguna titulación de alguna universidad.

  - Especialidad: {optativa, troncal, humanidades}

  - Formas Normales:

- **1FN.**

  Si todos sus atributos, para cada tupla, toman un único valor del dominio simple subyacente, es decir, no hay grupos repetitivos.

- **2FN.**

  Todo atributo no clave debe ser un hecho acerca de la **totalidad** de la clave.

- ***3FN.***

  Todo atributo no clave debe ser un hecho **sólo** acerca de la clave (y de la totalidad de la clave).

  - Las tablas son las siguientes:

- **Titulación** (Nombre_titulación, Campus, Créditos)

- **Curso** (Nombre_titulación, Nombre_curso, Duración, Coste)

- **Asignatura** (Nombre_Asignatura, Especialidad)

- **Posee** (Nombre_titulación, Nombre_curso, Nombre_Asignatura)

- **PaginaWeb** (Nombre_Asignatura, páginaweb)

- **Convalidaciones** (Nombre_Asignatura, Nombre_Asignatura_convalida)

- **Grupo** (Nombre_grupo, Criterio_evaluación, curso_academico)

- **Tiene** (Nombre_titulación, Nombre_curso, Nombre_grupo)
- **Hay** (Nombre_grupo, Nombre_Asignatura, horario)
- **Teoría** (Nombre_asignatura, Tema)
- **Práctica** (Nombre_asignatura, Nombre_práctica, fecha_entrega_practica*, fecha_inicio_practica)
- **Profesor** (Nombre_Profesor, Titulo)
- **Despacho** (Nombre_Profesor, Despacho, Teléfono)
- **Email** (Nombre_Profesor, Email)
- **Enseña_teoría** (Nombre_Profesor, Tema, Nombre_asignatura, Tutoría)
- **Enseña_practica** (Nombre_Profesor, Nombre_asignatura, Nombre_práctica, Tutoría)
- **TieneTeoria** (ID_Examen, Nombre_grupo, Tema)
- **TienePractica** (ID_Examen, Nombre_grupo, Nombre_práctica)
- **Examen** (ID_Examen, Día, Año, Hora, Fecha_Revisión)
- **TipoExamen** (id_examen, tipo)
- **AulaExamen** (id_examen, aula)
- **AulaTeoria** (nombre_grupo, tema, aula)
- **AulaPractica** (nombre_asignatura, nombre_practica, aula)
- **DiasPractica** (nombre_asignatura, nombre_practica, dia)
- **Esta** (Nombre_grupo, Nombre_profesor)

6.11.3. Pruebas de Bot que contesta a preguntas

Vamos a realizar las pruebas del bot que contesta a preguntas.

- Introducimos en el chat la siguiente pregunta:

  ¿Quería consultar los horarios de la asignatura procesadores del lenguaje que tengo este año en el grado de ingeniería informática?

- Vemos que el bot nos responde perfectamente tanto por voz como por texto:



- Introducimos en el chat la siguiente pregunta:

  ¿Que asignaturas hay en el grado de ingeniería informática?

- Vemos que el bot nos responde perfectamente tanto por voz como por texto:



- Introducimos en el chat la siguiente pregunta:

  ¿A qué hora son las tutorías de David Griol en Procesadores de Lenguaje?

- Vemos que el bot nos responde perfectamente tanto por voz como por texto:



- Introducimos en el chat la siguiente pregunta:

  ¿Donde está el despacho del Dr. David Griol?

- Vemos que el bot nos responde perfectamente tanto por voz como por texto:



- Introducimos en el chat la siguiente pregunta:

  ¿Cuando es el examen de Álgebra?

- Vemos que el bot nos responde perfectamente tanto por voz como por texto:



Conclusiones bot conversacional

El desarrollo de las redes sociales y los mundos virtuales ofrece una amplia gama de oportunidades y nuevos canales de comunicación que se pueden incorporar a las interfaces tradicionales. En este capítulo hemos explicado cómo integrar sistemas de diálogo en los mundos virtuales para crear metabots conversacionales. Siguiendo esta propuesta se ha desarrollado un metabot que proporciona información académica en Second Life y que es capaz de interaccionar oralmente con los avatares de los usuarios.

Second Life ofrece un gran número de posibilidades para la evaluación de estos nuevos canales y metodologías de comunicación dada la posibilidad que tienen sus usuarios de socializar, explorar, conocer a otros residentes y acceder a un gran número de recursos educativos y culturales.

Como trabajo futuro queremos evaluar nuevas características que pueden incorporarse al desarrollo del metabot conversacional y mejorar el proceso de comunicación. En especial, queremos llevar a cabo un estudio del efecto del comportamiento emocional del metabot en la comunicación oral, mediante la integración de texturas que permiten dotar de gestos y emociones a las expresiones faciales de los avatares.

CAPÍTULO VII

OPENSIM

Capítulo 7

OpenSim

7.1. Introducción

OpenSim es un simulador de código abierto que utiliza el mismo standard que Second Life para comunicarse con sus usuarios. Por ende hace posible que los usuarios puedan utilizar el mismo software que provee Linden Labs para conectarse a un servidor controlado por organizaciones o individuos sin relación con esta empresa.

OpenSim sirve para simular entornos virtuales de forma independiente al mundo de Second Life, desde una infraestructura propia.

OsGrid es la mayor red en la que corren OpenSimulator, y se puede conectar su propia región de OpenSimulator de forma totalmente gratuita.

En este capítulo explicamos:

- Cómo crear nuestra propia Sim con OpenSim en nuestro computador.
- Cómo crear el metaverso UC3M.
- Integrar Sloodle en Opensim.
- Física NINJA para OpenSim.

- Enlazar nuestra Sim de Opensim a OSGrid el metaverso OpenSource.

- Cargar una región remota.

- Hypergrid.

- Crear el Grid UC3M.

- Realxtend.

7.2. Descargar OpenSim

7.2.1. Descarga y Carpetas

Vamos a instalar OpenSim.

- Entramos en la página web de OpenSim:

  http://opensimulator.org/wiki/Main_Page

- Hacemos clic en Download Binaries:

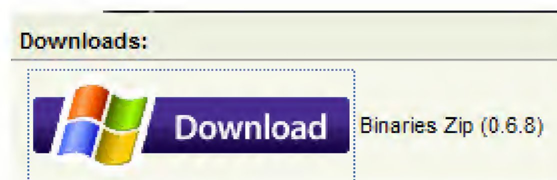
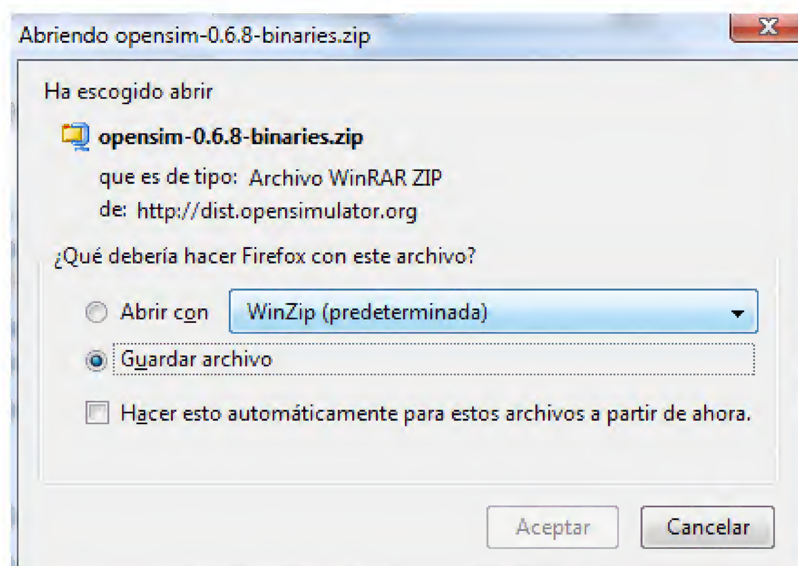
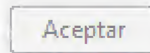


Figura 7.1. Descarga de OpenSim

- Nos sale el siguiente cuadro de dialogo:



- Hacemos clic en **Aceptar**:



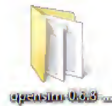
Elegimos donde guardar el fichero, yo por comodidad lo voy a copiar en la carpeta raíz c:.

- Nos crea el siguiente fichero:

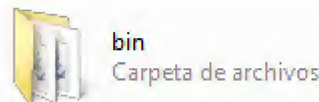


Descomprimimos el fichero con cualquier descompresor.

- Tras la descompresión obtenemos la siguiente carpeta y entramos en ella:



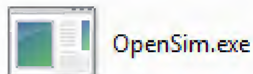
- Entramos en la carpeta bin:



7.3. Configuración de OpenSim

Vamos a configurar OpenSim.

- Para empezar la configuración hacemos doble clic en **OpenSim.exe**:



- Se abre una ventana que se denomina consola de configuración:

```
C:\opensim-0.6.8-binaries\bin\OpenSim.exe
00:07:36 - [INVENTORY DB]: Sqlite - connecting: URI=file:inventoryStore.db,versi
on=3
00:07:36 - [MIGRATIONS]: InventoryStore up to date, no migrations to apply
00:07:36 - [INVENTORY DB]: Populated Inventory Folders Definitions
00:07:36 - [INVENTORY DB]: Populated Inventory Items Definitions
00:07:36 - [INVENTORY SERVICE]: Initialized.
00:07:36 - [INVENTORY CONNECTOR]: Local inventory connector enabled
00:07:36 - [NEIGHBOUR CONNECTOR]: Local neighbour connector enabled
00:07:36 - [LAND CONNECTOR]: Local land connector enabled
00:07:36 - [REST COMMS]: Enabling InterregionComms RESTComms module
00:07:36 - [GRID SERVICE]: Starting...
00:07:36 - [LOCAL GRID CONNECTOR]: Local grid connector enabled
00:07:36 - [CLIENT]: LLProxyLoginModule port 0
00:07:36 - [LOADREGIONS]: Loading region configurations from filesystem
00:07:36 - [LOADREGIONSPLUGIN]: Loading region configurations...
=====================================

We are now going to ask a couple of questions about your region.

You can press 'enter' without typing anything to use the default

the default is displayed between [ ]: brackets.
=====================================

New region name []:
```

Figura 188. OpenSim.exe

En esta consola vemos como se va configurando nuestro OpenSim, y es el medio por el que, introduciendo los comandos adecuados, podemos administrarlo todo. Darte alta a usuarios, crear nuevas regiones, exportar, importar, etc.

Cuando acaban de ejecutarse los comandos de configuración básica, vemos que la consola se para en una línea de texto que indica **Región Name** y que por defecto pone **OpenSim Test**:

- Hacemos clic en el icono inicio:



- Abrimos el interprete de comandos, introduciendo la palabra **cmd**:



Abrimos el intérprete de comandos de Windows porque es más seguro arrancar OpenSim de esta manera en caso de que se produzca algún error.

- Se nos abre la consola:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  Reservados todos los derechos.

C:\Users\100048464>
```

- Nos colocamos en la carpeta bin de openSim:

```
C:\>cd opensim-0.6.8-binaries/bin/
```

- Ejecutamos OpenSim:

```
C:\opensim-0.6.8-binaries\bin>opensim
```

- Nos pide el nombre de la región y ponemos tesis:

```
New region name []: tesis
```

- Nos pide la UUID de la región y damos a la tecla enter:

```
Region UUID [53a5a292-056d-4ed5-a4b5-5e61d42a0d0b]:
```

- Nos pide la localización de la región y damos a la tecla enter:

```
Region Location [1000,1000]:
```

- Nos pide la dirección IP interna, y damos a la tecla enter:

```
Internal IP address [0.0.0.0]:
```

- Nos pide el puerto interno, y damos a la tecla enter:

```
Internal port [9000]:
```

- Nos dice si permitimos un puerto alternativo, y damos a la tecla enter:

```
Allow alternate ports [False]:
```

- Nos pide el nombre del host externo, y le damos la dirección **127.0.0.1**:

```
External host name [SYSTEMIP]: 127.0.0.1
```

- Nos pide la UUID del avatar Master, y damos a la tecla enter:

```
Master Avatar UUID [00000000-0000-0000-0000-000000000000]:
```

- Nos pide el primer nombre del avatar master, y introduzco el nombre de giaa:

```
Master Avatar first name (enter for no master avatar) []: giaa
```

- Nos pide el apellido de avatar master, y introduzo el apellido de Quixote:

```
Master Avatar last name []: Quixote
```

- Nos pide el Password de nuestro avatar master:

```
Master Avatar sandbox password []:
```

- Ahora vemos que ya se ha creado nuestra región tesis:

```
Region (tesis) #
```

No debemos de cerrar esta consola mientras que entremos en nuestra Sim de OpenSim.

- Abrimos la carpeta de **Regions**:



- Vemos que tenemos un archivo llamado Regions.ini y lo abrimos:



- Vemos el contenido del fichero Regions.ini:

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

- El archivo Regions.ini tiene los siguiente elementos:

  - **[tesis]:**

    Es el nombre de la región que le hemos dado.

  - **RegionUUID:**

    Es una identificación única para cada región.

  - **Location:**

    Son las coordenadas x e y donde está localizada nuestra región.

  - **InternalAddress:**

    La dirección IP interna es por defecto 0.0.0.0.

  - **InternalPort:**

    Es el puerto por el que nos comunicamos con la región, que es el 9000.

  - **Master_avatar_first, master_avatar_last y master_avatar_password:**

    Los nombres, apellidos y claves de los avatares que pueden acceder al Sim.

7.3.1. Almacenar la información de OpenSim en una base de datos

Hay que guardar la información, para que cuando salgas del servidor y vuelvas a entrar, lo que tenias en el servidor no se pierda, y el aspecto del avatar tampoco se pierda.

OpenSim soporta las siguientes bases de datos:

- **SQLite**. (por defecto)

  Es una base de datos que viene incluida con OpenSim y se puede usar sin necesidad de configuración adicional. Es sobre todo para la intención de puesta en funcionamiento rápidamente, no para uso en producción.

- **MySQL**. (completamente soportada)

  Esta es la base de datos recomendado para cualquier uso más allá de la experimentación o pequeñas aplicaciones independientes.

- **MSSQL** (parcialmente soportada)

  Vamos a mirar la configuración de OpenSim.ini.

- Abrimos el fichero OpenSim.ini:
- Entramos en la sección **StandAlone**:

```
[StandAlone]
; Inventory database provider
inventory_plugin = "OpenSim.Data.SQLite.dll"
; inventory_plugin = "OpenSim.Data.MySQL.dll"
```

Figura 189. La base de datos SQLite de OpenSim

Podemos comprobar que la base de datos seleccionada es SQLite.

7.4. Accediendo a nuestra Región

7.4.1. Configurar navegador

Si utilizamos el navegador de Linden Lab en una Sim de OpenSim, el avatar se puede ver como una nueve. Esto es debido a que la empresa Lindel Lab cambia de versión de explorador de Second Life, y esto desactualiza las librerías opengl, para solucionar este problema podemos bajarnos otro navegador.

Para conectar tu nuevo Sim con tu usuario, conecta un visor Second Life con la siguiente línea de comando:

- Un cliente en una máquina como OpenSim:

  -loginuri http://127.0.0.1:9000/

➢ Un cliente en una misma LAN como OpenSim:

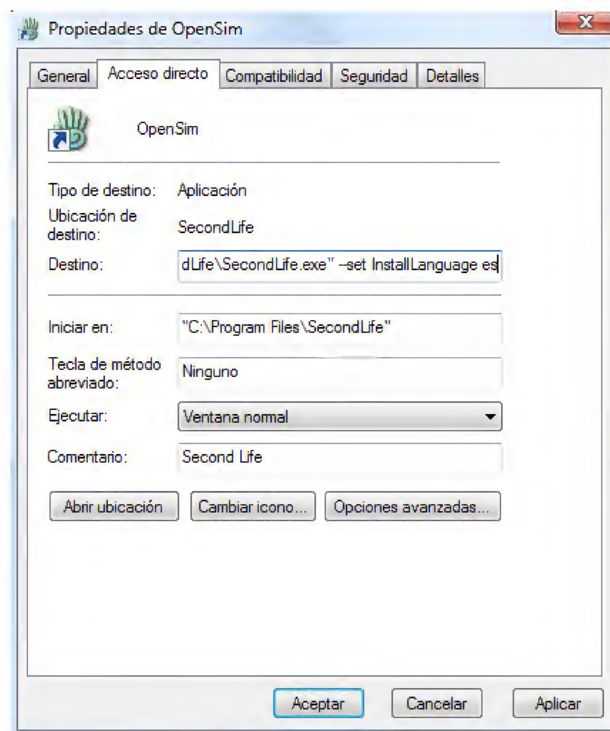
-loginuri http://lan_ip:9000/

➢ Un cliente en una diferente máquina o internet:

-loginuri http://external_ip:9000/

- Vamos al navegador de Second Life:

- Hacemos clic con el botón derecho del ratón en el acceso directo de Second Life y accedemos a sus propiedades:



- En el campo de Destino añadimos al final el texto (sin las comillas) "-loginuri http://127.0.0.1:9000".

Destino: tallLanguage es -loginuri http://127.0.0.1:9000

Figura 190. Loginuri

Le decimos mediante el comando loginuri que acceda a nuestra propia máquina, pues utilizamos la dirección 127.0.0.1 y que vamos a acceder al OpenSim de nuestra máquina por el puerto 9000. Por lo tanto siempre debemos de tener una región accesible con este puerto.

7.4.2. Acceder a nuestra a nuestra región OpenSim

Vamos a acceder a nuestra región OpenSim.

- Hay que dejar la consola abierta:

```
14:41:44 - [!]:STARTUP COMPLETE
Currently selected region is TESIS
14:41:44 - [STARTUP]: Startup took 12m 34s
Region (TESIS) #
```

Figura 191. Consola de la región Tesis

Esta consola no la debemos de cerrar si queremos entrar en OpenSim.

- Hacemos doble clic en el explorador de Second Life:



- Introducimos el nombre del avatar de nuestra región:

Nombre:
giaa

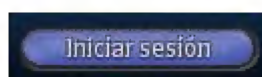
- Introducimos el apellido del avatar de nuestra región:

Apellido:
Quixote

- Introducimos la clave que hemos dado al avatar de nuestra región:

Contraseña:
****
Recordar la contraseña

- Hacemos clic en el botón Iniciar sesión:



- Accedemos a nuestra región como root en OpenSim:



Figura 192. Importar objetos de Second Life a OpenSim

Para importar objetos de Second Life a OpenSim, hay que tener permisos completos de los objetos (copiar, modificar y transferir).

7.5. Crear otro usuario en nuestras Sim de OpenSim

Vamos a crear otro usuario para nuestra Sim de OpenSim.

- Vamos a nuestra consola de comandos y escribimos **create user**:

```
Region (tesis) # create user
```

- Introducimos el nombre del avatar que queremos crear:

```
First name [Default]: saga
```

- Introducimos el apellido del avatar que queremos crear:

```
Last name [User]: Bayn
```

- Introducimos la clave que queremos dar a nuestro avatar:

```
Password
```

- Introducimos las coordenadas de la región donde queremos que se conecte nuestro avatar:

```
Start Region X [1000]:
Start Region Y [1000]:
```

- Introducimos la dirección e-mail de nuestro avatar:

```
Email []: e100048464@alumnos.uc3m.es
```

- Nos conectamos a nuestro OpenSim con nuestro nuevo avatar, Saga Bayn:



7.5.1. Entrar dos avatares a nuestra Sim de OpenSim

Vamos a entrar en nuestro Sim de OpenSim con los dos avatares, Saga Bayn y giaa Quixote.

- Vamos a utilizar en los dos avatares el mismo puerto en nuestro visor:

Destino: :allLanguage es -loginuri http://127.0.0.1:9000

- Abrimos dos visores de Linden:



- Introducimos los datos de nuestros dos avatares:



Figura 7.7. Dos avatares en nuestra Sim

Podemos apreciar que están conectados nuestros dos avatares en nuestra Sim de OpenSim en la misma computadora.

7.5.2. Conexión del bot a OpenSim

Vamos a conectar el bot a OpenSim.

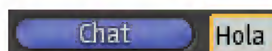
- Introducimos la siguiente línea de código fuente:

```
Client.Settings.LOGIN_SERVER = "http://127.0.0.1:9000";
```

* Vemos como se ha conectado nuestro bot a nuestra Sim de OpenSim:



* Introducimos un Hola en el chat:



* El bot nos responde perfectamente con Hola Sr. Saga Bayn:



7.6. Física NINJA para OpenSim

Este proyecto propone una extensión de la nueva física llamada NINJA Física. La extensión NINJA Física mejora la física ODE en OpenSim con articulaciones: vínculos movibles entre los prims. Esto permitirá que los vehículos de ruedas, muñecas de trapo, y mecánicos / robot montajes paras ser interactivas construcciones con el visor estándar y controlables a través de secuencias de script.

NINJA representa una red interactiva articulada de montajes. "Articulado de montajes" se refiere a un cuerpo rígido articulado.

7.6.1. Documentación del usuario

- Descarga el demo-playground.tgz:

  http://forge.opensimulator.org/gf/download/frsrelease/142/304/demo-playground.tgz

- Ya tenemos el **demo-playground.tgz**:



- Copia el **demo-playground.tgz** a tu opensim/**bin** directorio:

- Abrimos el fichero OpenSim.ini:



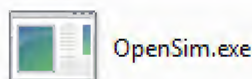
- Vemos el apartado NINJA Physics y vemos que esta desabilitada (esta comentada):

```
    ; if you would like physics joints to be enabled through a
special naming convention in the client, set this to true.
    ; (see NINJA Physics documentation,
http://opensimulator.org/wiki/NINJA_Physics)
    ; default is false
    ;use_NINJA_physics_joints = true
```

- Editar opensim.ini para ejecutar en modo standalone con la física ODE:

  use_NINJA_physics_joints Set = true.

- Arranca OpenSim.exe:



- En la región de la consola escribe **load-oar demo-playground.tgz** para cargar los datos demo:

```
Region (uc3m) # load oar demo-playground.tgz
```

- Hacemos clic en el cliente de OpenSim:



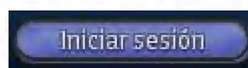
- Tenemos configurado el cliente de OpenSim para entrar en nuestra región:

Destino: stallLanguage es -loginuri http://127.0.0.1:900

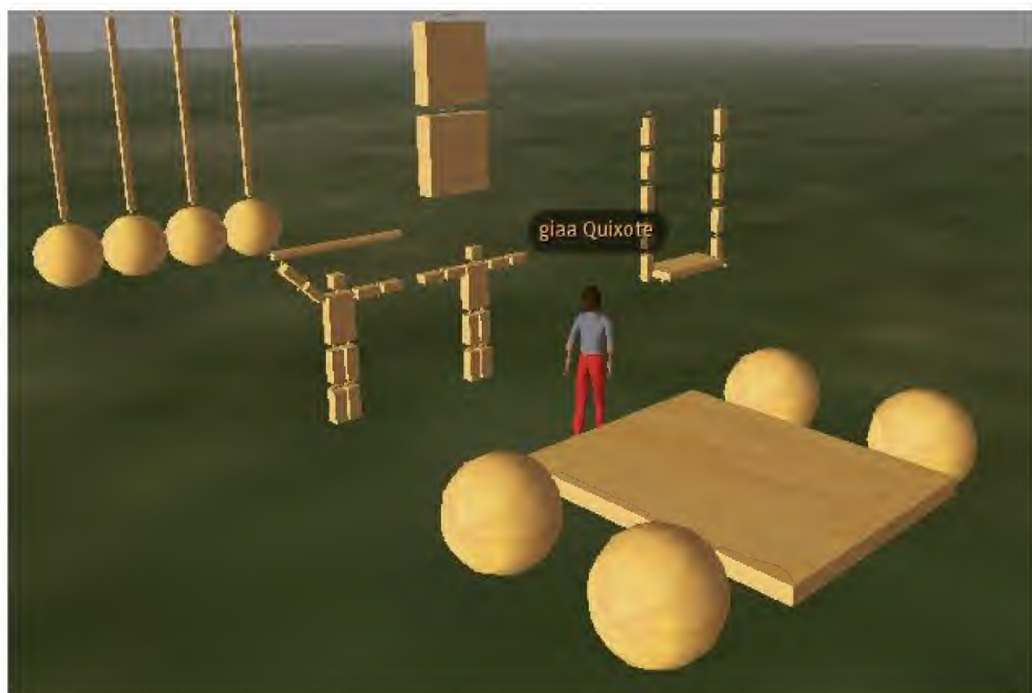
- Introducimos los datos de nuestro avatar:

Nombre:	Apellido:	Contraseña:
giaa	Quixote	*
		Recordar la contraseña

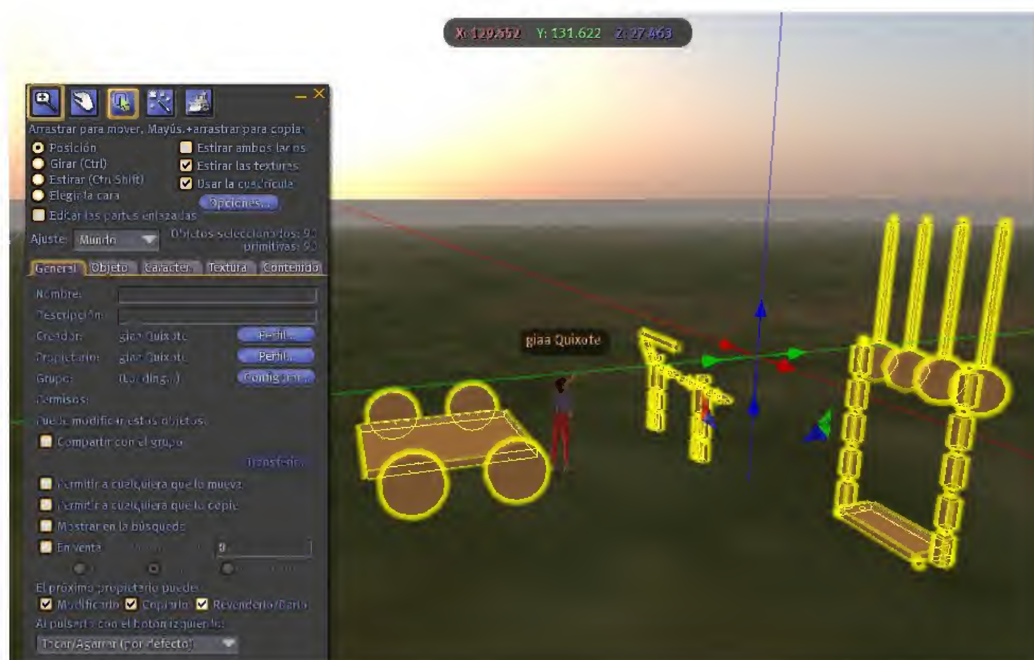
- Damos al botón **iniciar sesión**:



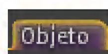
- Navega a las coordenadas (128.128):



- Seleccionar todos los prims en el objeto de oscilación (Ctrl +3, arrastre el ratón):



- En la ventana de edición de objeto, haga clic en la "física" en la casilla "Objeto":



- Ficha a su vez la selección Material:



- A continuación, presione ESC para salir del modo de selección. El prims y las articulaciones ir "en vivo".



- Haga clic en el montaje articulado y arrastrarlo alrededor.

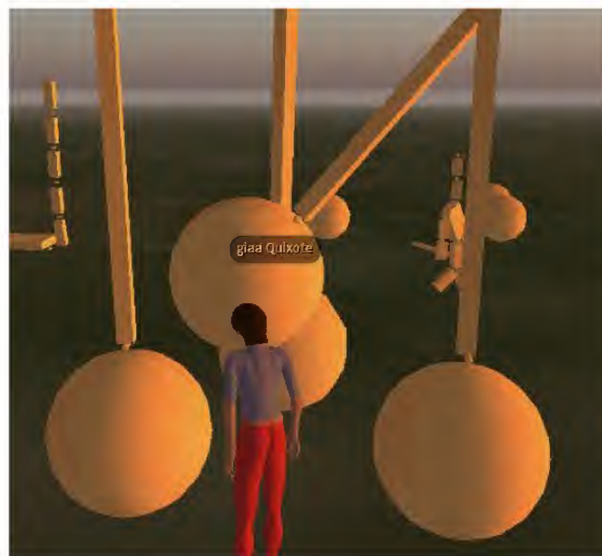
- Vemos como funcionan perfectamente las articulaciones del columpio:



- Nos columpiamos en el columpio:



- Vemos lo como se articula el péndulo:

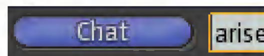


- Vemos el muñeco de trapo:



- Con los siguientes comandos de voz en el chat puede controlar el muñeco de trapo: "**arise**", "**kneel**", "**die**".

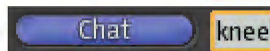
- Introducimos en el chat de voz la palabra **arise**, y damos a la tecla enter:



- Vemos como el muñeco de trapo se levanta:



- Introducimos en el chat de voz la palabra **kneel**, y damos a la tecla enter:



- Y vemos como el muñeco de trapo se arrodilla:



Figura 7.8. Muñeco de trapo

Me parece que tiene mucho futuro el poder cambiar cualquier estructura (edificio, construcción) sólo mediante comandos de voz.

7.6.2. Creación de una articulación bola y cavidad

Vamos a hacer un ejemplo de cómo utilizar la materia NINJA.

- Abra el archivo OpenSim.ini de la carpeta bin:

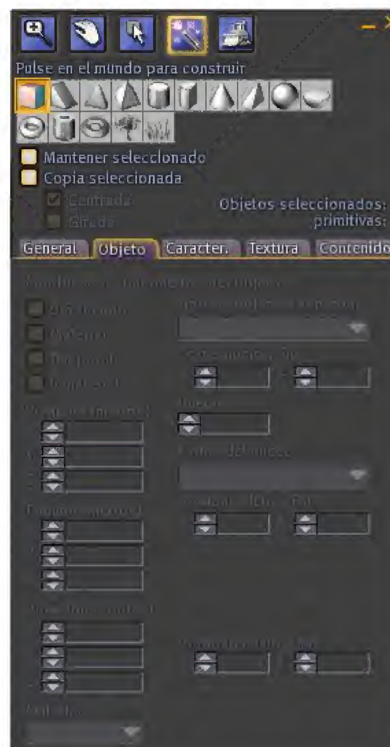


- En OpenSim.ini, configura "physics = OpenDynamicsEngine" y "use_NINJA_physics_joints = true".

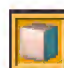
- Hacemos clic en el botón **Construir**:



- Se nos abre el siguiente cuadro:



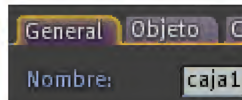
- Seleccionamos el prim tipo caja:

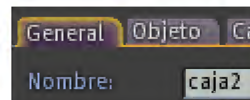


- Crea dos prim de tipo caja, lado con lado con espacio entre ellos. ¡Dejen los prims que estén como no materia, por ahora:



- Dé a cada uno un nombre único (seleccione el objeto, clic botón derecho del ratón, editar, pestaña General, Nombre): Ejemplos de nombres son "caja1" y "caja2".





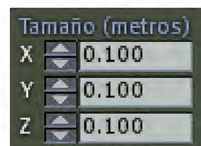
- Deja a las dimensiones (0,5, 0,5, 0,5):



- Estos dos prims estarán conectados por una articulación de rótula. Crear una pequeña esfera.



- Realiza sus dimensiones como algo (0,1, 0,1, 0,1) (seleccione el objeto, clic botón derecho del ratón, editar, pestaña General, Nombre):



- Esta prim representa una articulación de rótula y cavidad:



- Posición de la esfera entre las 2 cajas:



- Asignar la esfera pequeña el nombre de " balljoint-NombreÚnico, donde" NombreÚnico" es un nombre único, no usado por ningún otro prim.

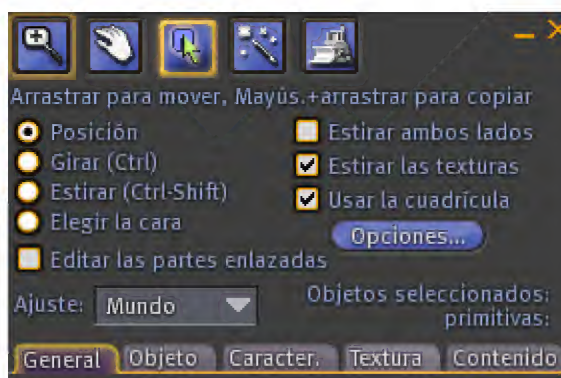
Nombre: balljoint-caja12

- Da a la esfera una descripción de "caja1 caja2", que son los nombres de los 2 prims que estar conectado, separadas por un espacio.

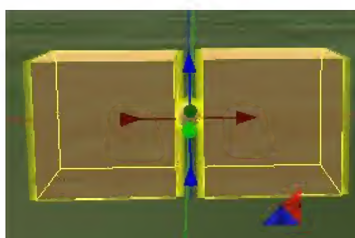
Descripción: caja1 caja2

- Cierre todas las ventanas en el visor.

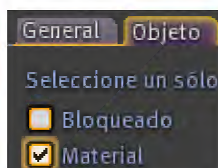
- Presione Ctrl +3 para entrar en modo de edición:



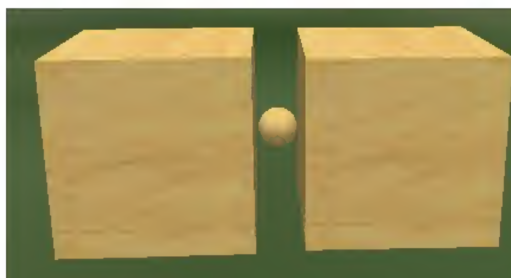
- Arrastre el ratón para marcar el 2 cajas para estar conectado y la pequeña esfera:



- En la ventana de edición de objeto, haga clic en la "materia" en la casilla "Objeto" ficha.



- Vemos cuadro y la bola:



- Arrastre los cuadros de alrededor (mantener pulsado el botón izquierdo del ratón sobre alguna caja a arrastrar) .

- Observa que se mantienen conectados por una articulación:



7.7. Sloodle en OpenSim

Vamos a subir las herramientas de Sloodle a nuestra Sim de OpenSim, de esta manera conectaremos nuestro Moodle a nuestra Sim de OpenSim, y de esta manera conseguiremos tras conectar nuestra Sim de OpenSim con Sloodle a OsGrid todo lo que necesitamos para crear varias aulas de enseñanza en un metaverso de forma totalmente gratuita.

- Vamos a la siguiente dirección web:

http://www.sloodle.org/downloads/sloodle-set-1.0.oar

Lo que vamos a descargar es un OAR en OpenSim, que es un Sim (isla) que contiene el Set Sloodle.

- Nos sale la siguiente ventana:

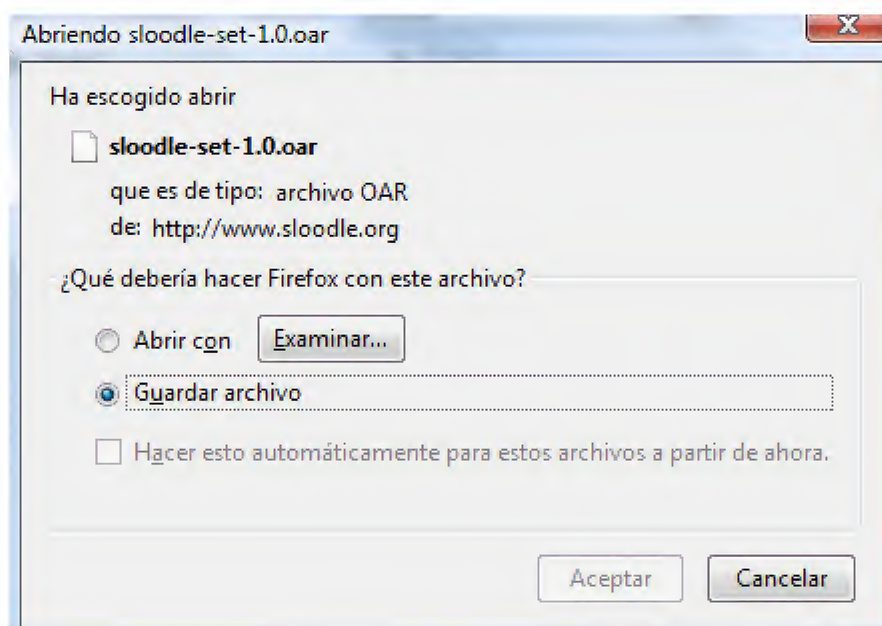


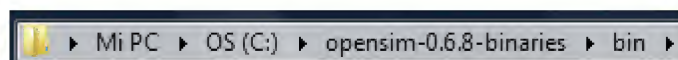
Figura 7.9. OAR OpenSim

- Hacemos clic en **Aceptar**:

- Nos crea el fichero sloodle-set-1.0:



- Copiamos el archivo sloodle-set-1.0.oar en la carpeta **bin** de nuestro OpenSim:



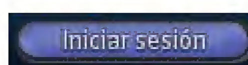
- Hacemos clic en nuestro navegador de Second Life adaptado para OpenSim:



- Introducimos los datos de nuestro avatar master en OpenSim:

Nombre: giaa | Apellido: Quixote | Contraseña: **************** | Iniciar sesión
Recordar la contraseña

- Hacemos clic en el botón **Iniciar sesión**:



- Entramos en nuestra Sim de OpenSim:



- Si tienes construido algo en tu isla tienes que poner **save oar [<nombre oar>]** sino perderás todo lo que tienes en tu isla.

- Vamos a cargar por consola con el comando **load oar[<nombre oar>]** el set de Sloodle:

```
Region (tesis) # load oar sloodle-set-1.0.oar
```

- Comprobamos que ya se ha cargado la oar del Sloodle set:

```
C:\Windows\system32\cmd.exe - opensim
Sim.sloodle_error_message_handler, script UUID 5a31826e-49b7-4191-8562-7ad28dd1e
be3, prim UUID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:17:58 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_mod_set-1.0, script UUID ca982b93-4983-464e-b1e6-6db6fb9a821f, prim
UUID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:17:58 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_set_effects_simple-1.0, script UUID 492a7a9c-6b3c-4f27-9f9f-2186c31e
0e74, prim UUID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:17:59 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_setup_notecard, script UUID b50cf31d-07c4-4fe0-aa27-157c2f52d78c, pr
im UUID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:17:59 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_setup_web, script UUID ecec6783-0e56-4132-9f75-381a0e4cfb07, prim UU
ID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:18:00 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_translation_en, script UUID 0afa65fc-124a-4b9a-9ef0-53573c39f53a, pr
im UUID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:18:00 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_translation_set_en, script UUID ad3d06ce-4d67-403a-aa88-2425d937caca
, prim UUID 34633730-7e37-423e-8cd7-8fae3ce28a2e @ <128, 128, 26.67591>
01:18:00 - [XEngine]: Loaded script SLOODLE 1.0 Sloodle Set Revision 13 for Open
Sim.sloodle_set_moodle_configured_effects-1.0, script UUID 59ab588e-c775-48b1-b2
19-38b6396da28e, prim UUID 444614d8-dc15-46f4-9bcd-04a40dc0a7db @ <128, 128, 26.
67591>
Region (tesis) # _
```

- Observamos que se ha cargado el set de Sloodle en nuestra OpenSim:



- Hacemos clic en el set de Sloodle:



7.7.1. Configuración del podio OpenSim

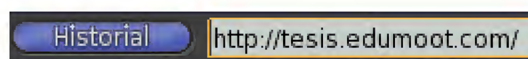
Vamos a configurar el podio del Set de Sloodle en OpenSim.

Recordemos que hay que dejar activo el curso el cual queramos utilizar con las herramientas de Sloodle.

- Hacemos clic en el podio de Sloodle, el cual esta esperando que introduzcamos la dirección web de nuestro sitio Moodle:

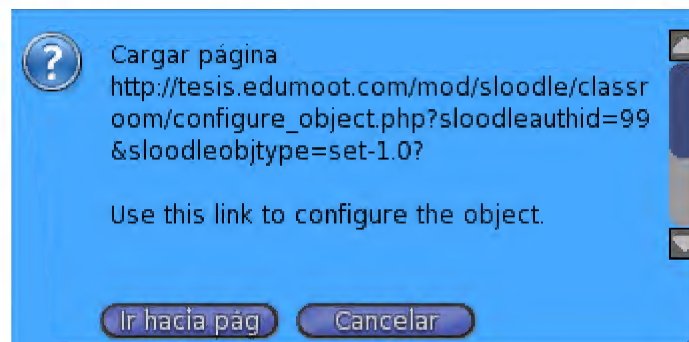


- Nos sale el siguiente mensaje para que introduzcamos la dirección de nuestro sitio Moodle:



- Introducimos la dirección de nuestro sitio Moodle:

- Nos aparece el siguiente cuadro azul:



- Hacemos clic en el botón **Ir hacia pag**:



- Nos aparece la siguiente página web para configurar el controlador de Sloodle:

Home » SLOODLE Module » SLOODLE Object Authorization

Object Details
Object Name: SLOODLE 1.0 Sloodle Set Revision 13 for OpenSim
Object UUID: 34633730-7e37-423e-8cd7-8fae3ce28a2e
Object Type: set-1.0

SLOODLE Object Authorization

Select Controller:

3. OpenSim y OsGrid
SLOODLE Controller: Controlador para OpenSim
9. Second Life
SLOODLE Controller: Curso Second Life

Do you want to authorize this object? Yes

Submit

- Elegimos que queremos utilizar el controlador del curso de Sloodle:

9. Second Life
SLOODLE Controller: Curso Second Life

- Hacemos clic en el botón **Submit**:

Submit

- Nos aparece la siguiente página web informando de que la autorización del objeto ha sido exitosa:

Home » SLOODLE Module » Object Configuration

Authorizing for: Second Life > SLOODLE Controller: Curso Second Life

Object authorization has been successful.

Object Details
Object Name: SLOODLE 1.0 Sloodle Set Revision 13 for OpenSim
Object UUID: 34633730-7e37-423e-8cd7-8fae3ce28a2e
Object Type: set-1.0

Object Configuration

No additional configuration options for this object.

- Vamos al curso de Second Life de nuestro plataforma Moodle:

Second Life





- Hacemos clic en el enlace del controlador de Sloodle del curso Second Life:

SLOODLE Controller: Curso Second Life

- Nos aparece la siguiente tabla informándonos de que el objeto Sloodle 1.0 Set Revision 13 for OpenSim esta autorizado:

Authorized Objects

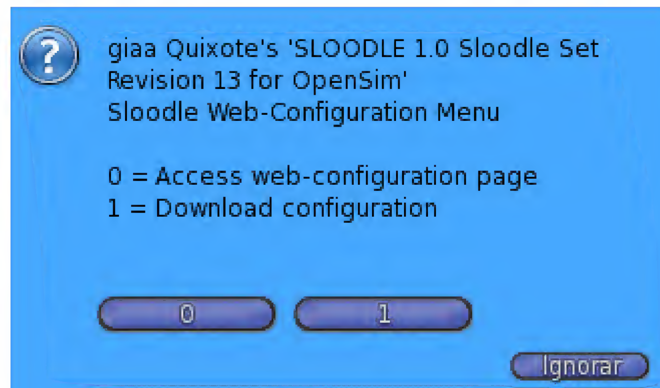
Object Name	Object UUID	Object Type	Last Updated	
SLOODLE 1.0 Sloodle Set Revision 13 for OpenSim	34633730-7e37-423e-8cd7-8fae3ce28a2e	set-1.0	2010-03-30 04:29:22 CDT	☐

Delete Selected

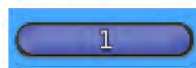
- Vemos que el Set Sloodle está esperando que lo configuremos, y para ello hacemos clic en él:



- Nos aparece el siguiente cuadro azul diciéndonos si queremos que lo configuremos o que bajemos la configuración que acabos de realizar:



- Hacemos clic en el botón 1, para que baje la configuración que acabamos de realizar:



- Ahora vemos que el Set de Sloodle acaba de recibir la configuración:



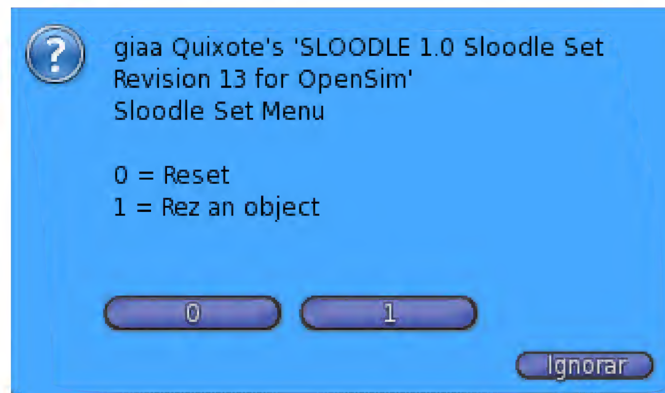
7.7.2. Configuración de MetaGloss en OpenSim

Vamos a configurar la herramienta MetaGloss en OpenSim.

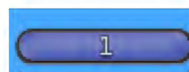
- Vemos que nuestro Set de Sloodle ya está configurado en nuestro curso de Second Life de nuestro sitio Moodle:



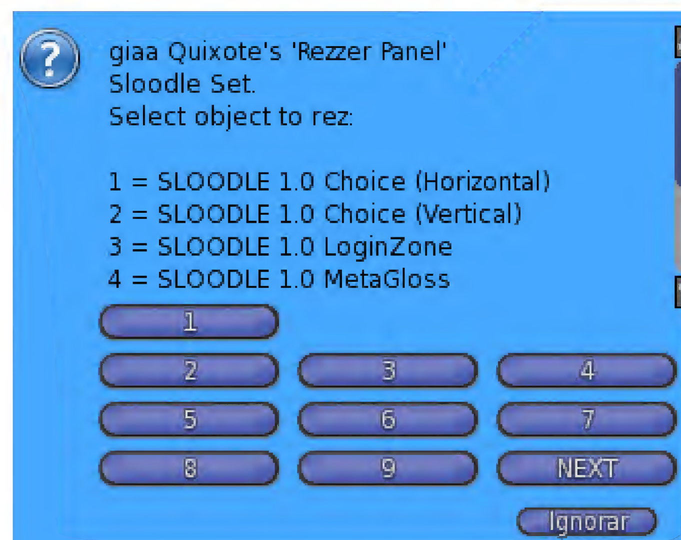
- Nos aparece la siguiente ventana azul indicándonos si queremos resetear la configuración del Sloodle Set o si queremos crear un objeto del Sloodle Set:

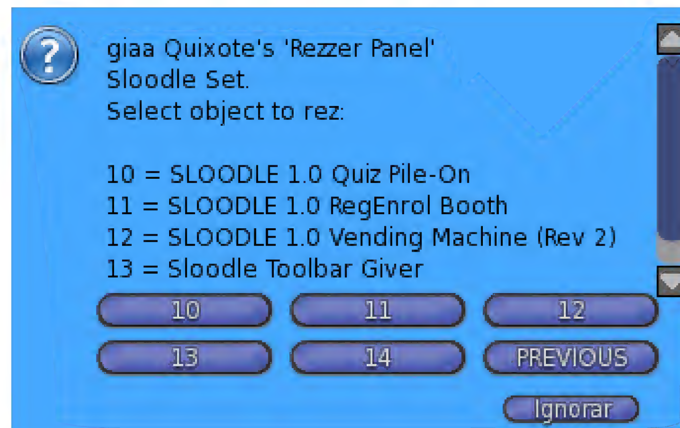


- Hacemos clic en el botón 1 para crear un objeto del Set de Sloodle:

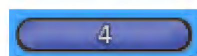


- Nos aparece los objetos que podemos crear:





- Hacemos clic en el botón **4**:



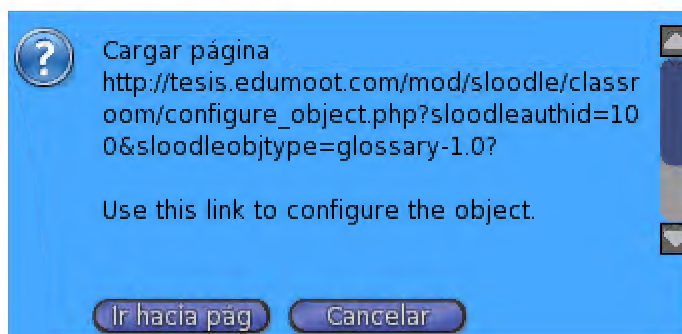
- Se nos crea en nuestra Sim el objeto metagloss del Set de Sloodle:



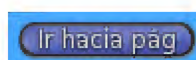
- Vemos que el objeto metagloss del Set de Sloodle está esperando bajar la configuración de nuestro sitio Moodle del curso Second Life:



- Nos sale el siguiente cuadro azul para cargar la configuración:



- Hacemos clic en el botón **Ir hacia pag**:



- Nos aparece la siguiente página web para elegir de nuestro sito Moodle del curso de Second Life, el controlador que queremos seleccionar y las opciones que podemos elegir para configurar:

Home • SLOODLE Module • Object Configuration

This object has already been authorized. If you want to re-authorize it, then please delete its authorization entry from your SLOODLE Controller.

Object Configuration

General Configuration

Select Glossary: Glosario de entrada

Show Partial Matches: Yes

Search Aliases: No

Search Definitions: No

Idle timeout (seconds): 120

Object Details

Object Name: SLOODLE 1.0 MetaGloss

Object UUID: 4271b23f-fd0a-4230-96d1-1883d5c46d20

Object Type: glossary-1.0

Access Level

Object Access Level

This determines who may access the object in-world

Use object: Public

Control object: Owner

Server Access Level

This determines who may use the server resource

Access Level: Public

Submit

- Hacemos clic en el botón **Submit**:

Submit

- Nos aparece la siguiente información de que ha sido autorizado el objeto de Metagloss de Sloodle 1.0 para el controlador del Curso Second Life:

Object Details

Object Name: SLOODLE 1.0 MetaGloss

Object UUID: 4271b23f-fd0a-4230-96d1-1883d5c46d20

Object Type: glossary-1.0

Authorized for: Second Life > SLOODLE Controller: Curso Second Life

Number of settings stored: 9

Continue

- Hacemos clic en el botón **Continue**:

Continue

- Ahora podemos apreciar como nuestro Meta Glossary ha recibido la configuración, y la configuración ha sido exitosa:



7.7.3. Prueba de MetaGloss en OpenSim

Vamos a realizar la prueba de la herramienta MetaGloss.

- Introducimos en el chat de OpenSim la palabra **/def Second Life**:

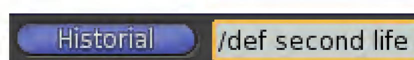


Figura 7.10. Def de metagloss

De esta manera conseguimos la definición del término Second Life que está en nuestro sitio Moodle.

- Vemos como el MetaGloss de OpenSim empieza la búsqueda:



- Vemos que el MetaGloss nos ha dado la definición de la palabra Second Life que aparece en nuestro sitio Moodle:



7.7.4. Ver video en Sloodle Presenter

Vamos a ver que funciona perfectamente el video en nuestra Sim.

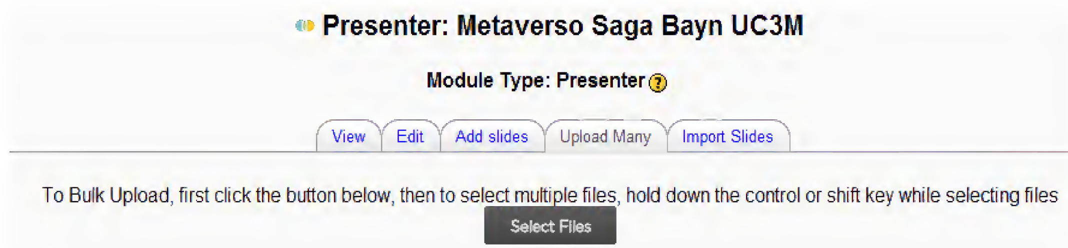
- Hacemos clic en el controlador Presenter: Metaverso Saga Bayn UC3M, que hemos creado previamente, y anteriormente en la memoria se cita como hacerlo:

Presenter: Metaverso Saga Bayn UC3M

- Seleccionamos la pestaña **Upload Many**:

Upload Many

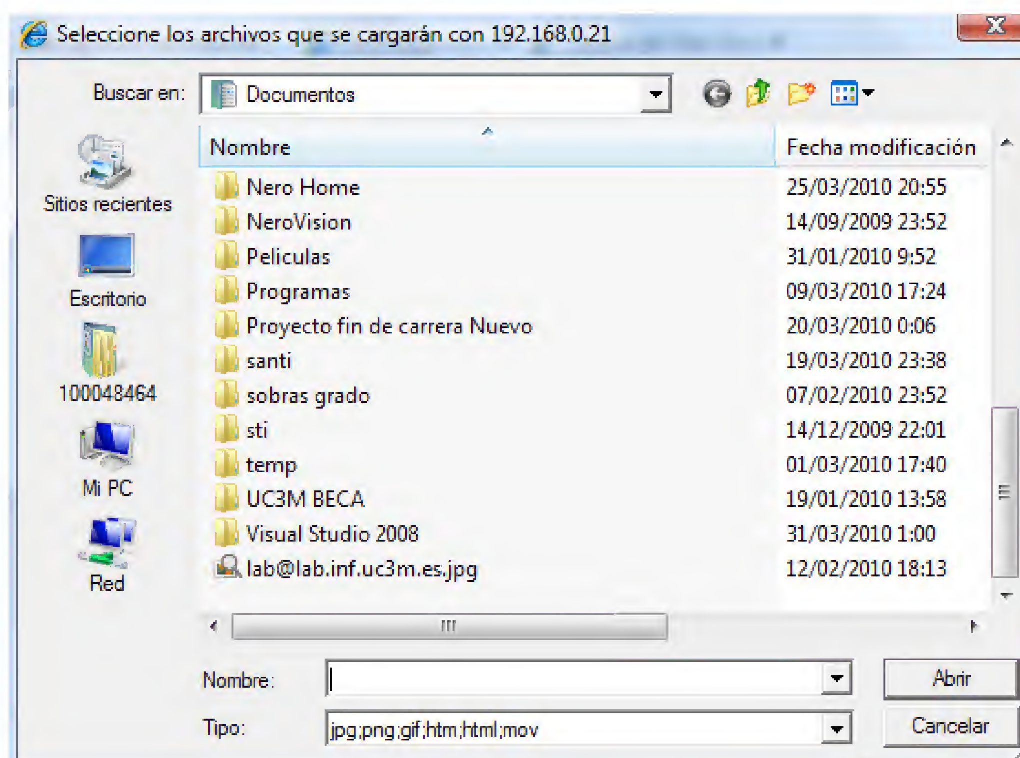
- Nos aparece la siguiente ventana:



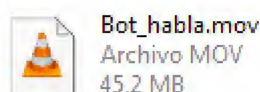
- Hacemos clic en el botón **Select Files**:

Select Files

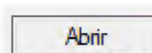
- Se nos abre el cuadro de dialogo para subir archivos:



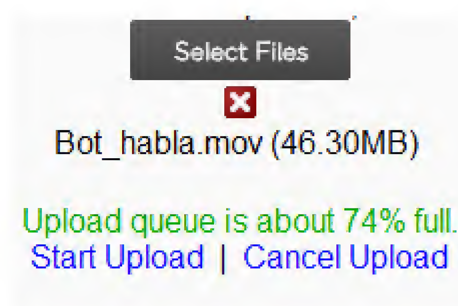
- Seleccionamos el archivo Bot_habla, que esta grabado en formato mov, que es el formato que acepta Moodle:



- Hacemos clic en el botón abrir:



- Nos aparece la siguiente información indicándonos el tamaño del fichero Bot_habla, y dándonos la oportunidad de subir el fichero o cancelar la subida:



- Hacemos clic en el botón **Start Upload**:

Start Upload

- Nos aparece la siguiente información sobre el nombre del video y la dirección donde está colgado el video:

Name: Bot_habla Type: video

Url: http://192.168.0.21/file.php/1/presenter/349/Bot_habla.mov

Add the above files to the presentation

- Hacemos clic en el botón **Add the above file to the presentation**:

Add the above files to the presentation

- Vemos que se ha añadido la película Bot habla a nuestra presentación:

Position	Select All	Name	Type	Actions
1	☐	Universidad Carlos III de Madrid	Web	⇅ ✎ 🔍 X +
2	☐	Open Metaverse	Web	⇅ ✎ 🔍 X +
3	☐	Bot_habla	Video	⇅ ✎ 🔍 X +
	Select All			Delete Selected +

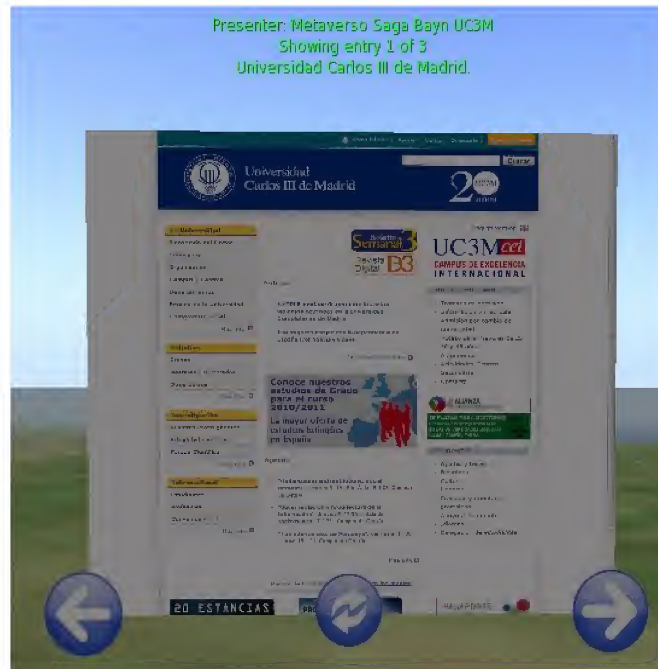
- Vamos utilizar para ello el Presenter de Sloodle, que anterior mente explicamos como funcionaba:



- Hacemos un clic en la flecha:



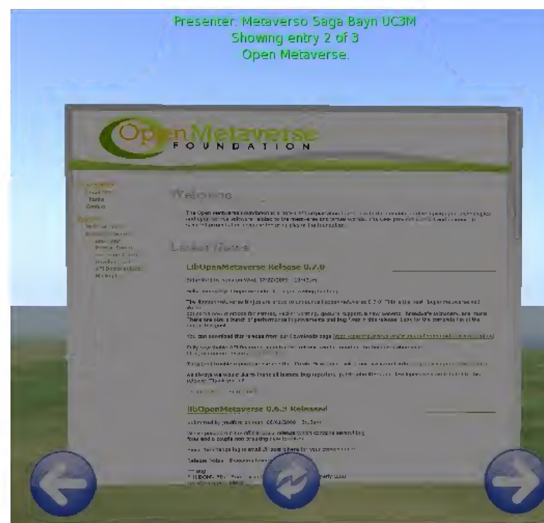
- Vemos la página web de la universidad:



- Hacemos un clic en la flecha:



- Ahora vemos la página web de OpenMetaverse:



- Hacemos un clic en la flecha:



- Ahora vemos el video del proyecto, se ve y se oye perfectamente:



7.8. Creando nuevas regiones

7.8.1. Creación de los UUIDs de las regiones que creemos

Vamos a crear tres regiones o islas más, para ello necesitamos conseguir los UUIDs ó identificadores únicos de estas regiones.

- Salimos de OpenSim escribiendo en la consola **quit** o escribir **shutdown**:

```
Region (Tesis) # quit
```

- Abrimos la carpeta de Regions:



- Vamos a la siguiente dirección web:

http://www.famkruithof.net/uuid/uuidgen

- Se nos abre la siguiente ventana:

UUID (GUID) Generator on the WEB - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://www.famkruithof.net/uuid/uuidgen

Comenzar a usar Fire... Últimas noticias IRENE MUNI en Seco... Cuentas de Google GLIntercept

The following version 1 UUID / GUID is generated for your use:

c60448c0-be21-11dc-95ff-0800200c9a66

This UUID is generated according to RFC 4122 Using the timestap / nodeid version (version 1), where the nodeid comes from network equipment I own.

Fetch multiple UUIDs at once: 1 (at most 500 at once)
Fetch UUIDs of another type: Version 1: Time/Node based Get

This type of UUID is generated using the current time, a clock id which changes in case the current time is found to be older than the latest known time a UUID is generated and an IEE 802 hardware address which should be unique. Still the following disclaimer applies:
Disclaimer:
The provided UUID (GUID) is provided AS IS without warranty of any kind, not even the warranty that the generated UUID is actually unique. The entire risk of using this UUID is upto you. If you cannot agree to those terms do not use the generated UUID.
Please do not use UUID from a cached paged.

Extract the time from a version 1 UUID

More information about UUIDs.

Contact information on the homepage of this host.

- En **Fetch multiple UUIDs at once** introducimos un 4:

UUID (GUID) Generator on the WEB - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://www.famkruithof.net/uuid/uuidgen

Comenzar a usar Fire... Últimas noticias IRENE MUNI en Seco... Cuentas de Google GLIntercept

The following version 1 UUIDs/GUIDs are generated for your use:

0186a8c0-be22-11dc-95ff-0800200c9a66
0186a8c1-be22-11dc-95ff-0800200c9a66
0186a8c2-be22-11dc-95ff-0800200c9a66
0186a8c3-be22-11dc-95ff-0800200c9a66

These UUIDs are generated according to RFC 4122 Using the timestap / nodeid version (version 1), where the nodeid comes from network equipment I own.

Fetch multiple UUIDs at once: 4 (at most 500 at once)
Fetch UUIDs of another type: Version 1: Time/Node based Get

Introducimos un 4 para que nos de las UUIDs para cada región. Cogemos 4 UUIDs para tener una UUID de reserva, tenemos el fichero "default" que ya tiene una UUID válida, por lo que sólo necesitamos 3 UUIDs más, pero es conveniente tener siempre una UUID de reserva.

7.8.2. Creación de tres regiones

Vamos a crear tres regiones para OpenSim.

- Vamos a la carpeta región:



- Hacemos clic en el fichero Regions.ini:

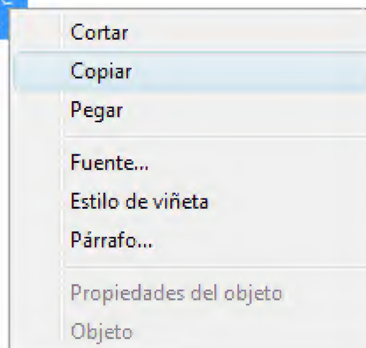


- Vemos nuestra región tesis:

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

- Deberemos copiar este contenido:

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```



- Ahora pegamos el contenido, debajo de la región tesis, este paso lo repetiremos 2 veces más para así crear 3 regiones:

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

- Y ahora editaremos el **nombre de la región**, la **RegionUUID**, la **Location**, el **IternalPort**. El **nombre**, **apellido** y **clave de acceso** los pondremos según nuestra elección.

- Las regiones tendrán los siguientes puertos y estarán en las coordenadas siguientes:

  - **La región por tesis:**

    Estará en las siguientes coordenadas 1000,1000, y usara el puerto 9000.

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

  - **La región UC3M:**

    Estará en las siguientes coordenadas 1000,1001, y usara el puerto 9001.

```
[uc3m]
RegionUUID = 0186a8c0-be22-11dc-95ff-0800200c9a66
Location = 1000,1001
InternalAddress = 0.0.0.0
InternalPort = 9001
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

- **La región UC3M 2:**

Estará en las siguientes coordenadas 1001, 1000 y usara el puerto 9002.

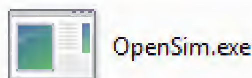
```
[uc3m 2]
RegionUUID = 0186a8c1-be22-11dc-95ff-0800200c9a66
Location = 1001,1000
InternalAddress = 0.0.0.0
InternalPort = 9002
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

- **La región UC3M 3:**

Estará en las siguientes coordenadas 1001, 1001 y usara el puerto 9003.

```
[uc3m 3]
RegionUUID = 0186a8c2-be22-11dc-95ff-0800200c9a66
Location = 1001,1001
InternalAddress = 0.0.0.0
InternalPort = 9003
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

- Hacemos clic en OpenSim.exe:



- Se nos abre la consola de comandos:

```
C:\opensim-0.6.8-binaries\bin\OpenSim.exe
ri: 8, State: Wait, Reason: ExecutionDelay
ID: 1056, TotalProcessorTime: 00:00:00, TimeRunning: 00:59:51.1000000, Pri: 8, S
tate: Wait, Reason: UserRequest
ID: 4572, TotalProcessorTime: 00:00:00.0624004, TimeRunning: 00:59:50.8880000, P
ri: 9, State: Wait, Reason: UserRequest
ID: 6060, TotalProcessorTime: 00:00:00.2496016, TimeRunning: 00:59:50.8880000, P
ri: 8, State: Wait, Reason: ExecutionDelay
ID: 3304, TotalProcessorTime: 00:00:00.0468003, TimeRunning: 00:59:50.8880000, P
ri: 9, State: Wait, Reason: ExecutionDelay
ID: 1160, TotalProcessorTime: 00:00:00, TimeRunning: 00:59:42.7810000, Pri: 8, S
tate: Wait, Reason: ExecutionDelay
ID: 2524, TotalProcessorTime: 00:00:00, TimeRunning: 00:59:41.0980000, Pri: 8, S
tate: Wait, Reason: ExecutionDelay
ID: 2684, TotalProcessorTime: 00:00:00.3276021, TimeRunning: 00:56:27.5680000, P
ri: 9, State: Wait, Reason: UserRequest
ID: 1036, TotalProcessorTime: 00:00:00.1248008, TimeRunning: 00:56:06.6130000, P
ri: 9, State: Wait, Reason: UserRequest
ID: 5744, TotalProcessorTime: 00:00:00, TimeRunning: 00:16:44.2620000, Pri: 9, S
tate: Wait, Reason: EventPairLow

*** ThreadPool threads ***
workers: 1 (500); ports: 0 (1000)


Region (root) #
```

- Hacemos clic en el navegador de Second Life que está adaptado para OpenSim:



- Introducimos los datos de nuestro avatar, y damos al botón iniciar sesión:

Nombre:	Apellido:	Contraseña:	
giaa	Quixote	****************	Iniciar sesión
		Recordar la contraseña	

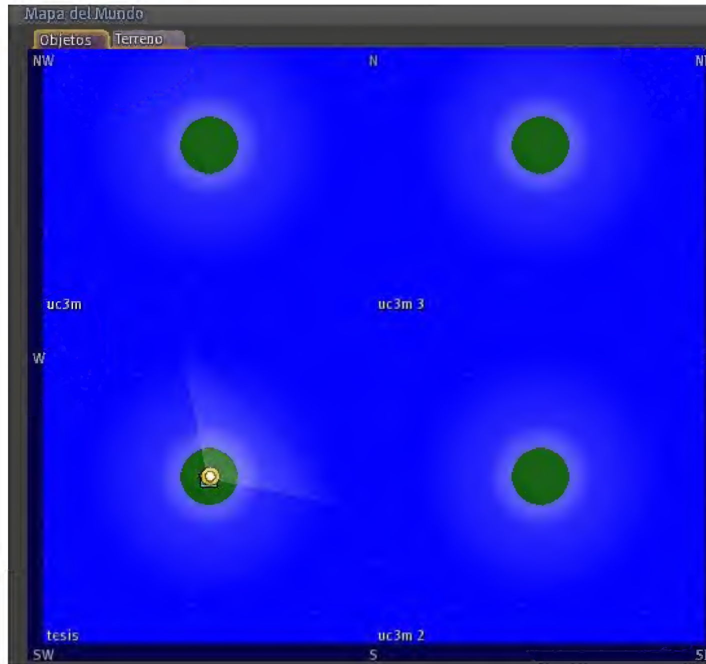
- Entramos en nuestra región tesis:



- Damos al botón Mapa:



- Vemos que tenemos cuatro Sim en el mapa:



- Para configurar nuestra isla en OpenSim existe una serie de comandos.
- Vamos a la siguiente dirección web para ver los comandos que existen:

  http://opensimulator.org/wiki/Console_Commands

  http://opensimulator.org/wiki/Server_Commands

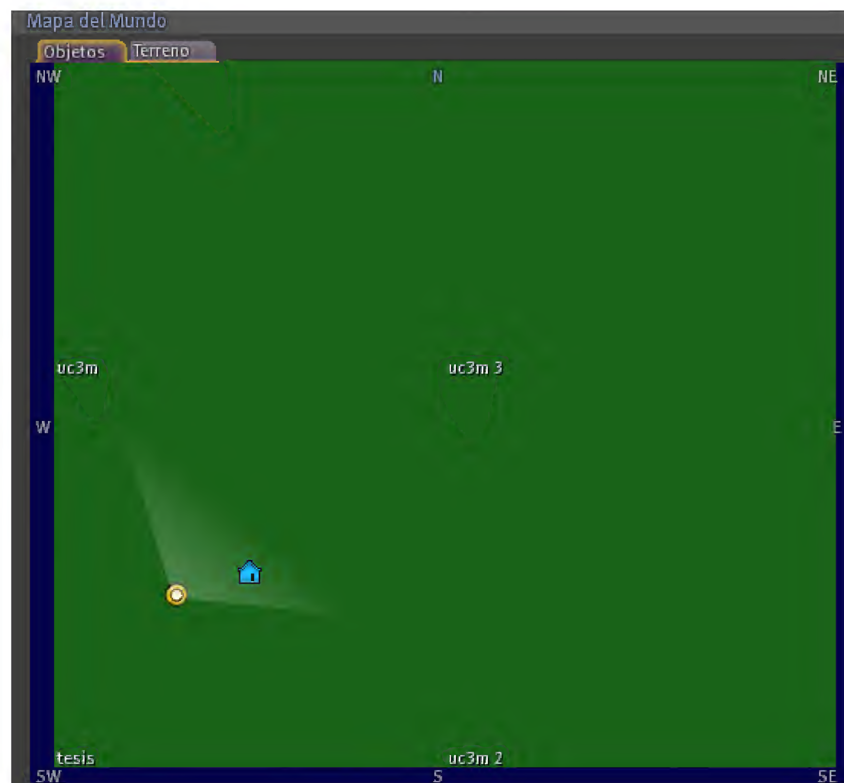
- Ahora alisamos las tesis con el comando terrain fill 25:

```
Region (root) # terrain fill 25
```

- Comprobamos que hemos alisado el terrano:



- Vemos como hemos alisado todas las islas:



7.8.3. Creamos una región de otro avatar

Vamos a crear la región de saga Bayn.

- Creamos el fichero ifonsa.ini:



- Ahora tenemos dos ficheros .ini:



- Editamos el fichero ifonsa.ini:

```
[ifonsa]
RegionUUID = 0186a8c3-be22-11dc-95ff-0800200c9a66
Location = 1000,1002
InternalAddress = 0.0.0.0
InternalPort = 9004
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = saga
MasterAvatarLastName = Bayn
MasterAvatarSandboxPassword =
```

Figura 7.11. Edición del fichero ifonsa.ini

Le damos una identificación de Región UUID, otra localización, y otro puerto, por supuesto decimos que el avatar propietario es saga Bayn.

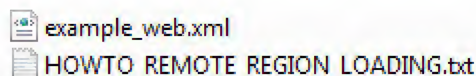
- Vemos que se a creado el Sim del avatar saga Bayn y que el avatar giaa Quixote que es el máster de OpenSim puede entrar sin problemas a la isla de saga Bayn:



7.9. Como cargar una región remota

La carga de una región remota habilita un fácil mantenimiento de las regiones que corren en un simulador desde un servidor Web.

- La información de cómo cargar una región está en la carpeta OpenSim/ share/ RegionLoading, la cual vemos dos fichero:

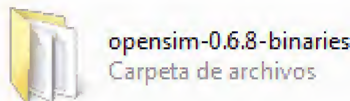


- El fichero **example_web.xml** es un ejemplo de como crear una versión web para cargar una región:

```
- <Regions>
  - <Root>
      <Config sim_UUID="18fb66dbf6274279885228f1c4064f8c" sim_name="OpenSim Test" sim_location_x="1000" sim_location_y="1000" datastore="OpenSim.db"
        internal_ip_address="0.0.0.0" internal_ip_port="9000" external_host_name="127.0.0.1" master_avatar_first="Test" master_avatar_last="User" master_avatar_pass="test" />
    </Root>
  - <Root>
      <Config sim_UUID="111111dbf6274219881228f1c1061f8c" sim_name="OpenSim Test2" sim_location_x="1001" sim_location_y="1000" datastore="OpenSim2.db"
        internal_ip_address="0.0.0.0" internal_ip_port="9001" external_host_name="127.0.0.1" master_avatar_first="Test" master_avatar_last="User" master_avatar_pass="test" />
    </Root>
  </Regions>
```

- Los pasos que vamos a seguir vienen en el fichero HOWTO_REMOTE_REGION_LOADING, es decir, como cargar una región remota.

- Vamos a la carpeta de OpenSim:



- Buscamos el fichero OpenSim.ini que está dentro de la carpeta OpenSim:



- Entramos dentro del fichero OpenSim.ini y vemos la siguiente configuración, para determinar donde OpenSimulator busca los ficheros donde le dicen las regiones para el servidor. Por defecto busca la información de las regiones en filesystem (en mi sistema de ficheros, es decir, en mi propio ordenador):

```
    ; ##
    ; ## REGIONS
    ; ##

    ; Determine where OpenSimulator looks for the files which tell it
which regions to server
    ; Defaults to "filesystem" if this setting isn't present
    region_info_source = "filesystem"
    ; region_info_source = "web"
```

- Cambiamos el 'region_info_source = filesystem' debajo de [Starup] to 'region_info_source = web':

```
; region_info_source = "filesystem"
  region_info_source = "web"
```

- Esta parte de la configuración determina la pagina desde las que son recibidas las regiones XML si tu las cargas las regiones desde la web. Los ficheros XML van entre <Root> </Root>, pero también te los puedes encontrar <Regions> <Root> <Root> <Regions>:

```
    ; Determines the page from which regions xml is retrieved if you
are loading these from the web
    ; The XML here has the same format as it does on the filesystem
(including the <Root> tag),
    ; except that everything is also enclosed in a <Regions> tag.
    ; regionload_webserver_url = "http://example.com/regions.xml";
```

- Cambia la línea 'regionload_webserver_url = ' a 'regionload_webserver_url = http://127.0.0.1/default.xml' remplazando http://127.0.0.1 con la URL del fichero de la región XML:

```
    ; Determines the page from which regions xml is retrieved if you
are loading these from the web
    ; The XML here has the same format as it does on the filesystem
(including the <Root> tag),
    ; except that everything is also enclosed in a <Regions> tag.
    regionload_webserver_url = "http://127.0.0.1/default.xml";
```

- El fichero XML de una remota región es similar al filesystem versión, excepto que es un archivo en lugar de múltiples archivos. Es decir, nosotros tenemos un archivo para cada isla, pues el fichero XML de una región remota utiliza un archivo donde están varias islas.

  Ejemplo:

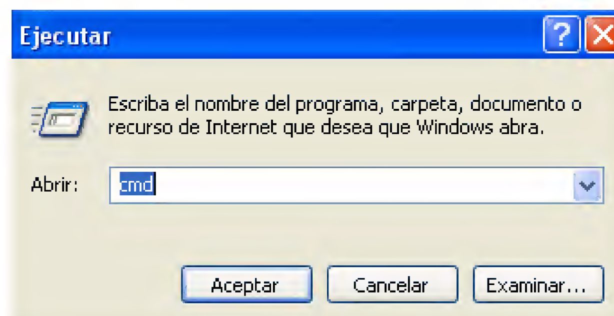
  Dentro del fichero example_web.xml tenemos <Regions> <Root> <Root> <Root> <Root> <Regions>, por lo tanto hay dos islas en un fichero.

7.10. Metaverso OpenSim

Vamos a crear el Metaverso de la Universidad Carlos III de Madrid.

- Para saber los datos de red de nuestro ordenador hacemos clic en el botón inicio/ejecutar:

- Dentro de la ventana ejecutar escribimos el comando **cmd**:



- Ponemos en la consola el siguiente comando **ipconfig**:

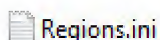
```
C:\Users\100048464>ipconfig

Configuración IP de Windows


Adaptador LAN inalámbrico Conexión de red inalámbrica:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::e43e:6c4f:decc:b2bc%12
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.0.21
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.0.1
```

- Vamos a nuestro fichero de Regions, que está dentro de la carpeta bin:



- Introducimos en el fichero regions en ExternalHostName nuestra dirección IP:

```
[tesis]
RegionUUID = 53a5a292-056d-4ed5-a4b5-5e61d42a0d0b
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 192.168.0.21
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword = granota09
```

- Conectamos la consola de OpenSim:

```
Region (tesis) #
```

- Configuramos el visor de Second Life:



- Configuramos el navegador, poniendo la nuestra dirección IP **192.168.0.21**:

Destino: anguage es -loginuri http://192.168.0.21:9000

- Introducimos los datos de nuestro avatar:

Nombre:	Apellido:	Contraseña:
saga	Bayn	********

- Hacemos clic en el botón **iniciar sesion**:



- Vemos que nos hemos conectado al metaverso de la Universidad Carlos III de Madrid que acabamos de crear:



- Introducimos esta línea de código en el código fuente de nuestro bot, que vale para que se conecta a nuestro metaverso de OpenSim, en la cual hemos introducido la dirección IP de nuestro metaverso:

```
Client.Settings.LOGIN_SERVER = "http://192.168.0.21:9000";
```

- Vemos que el bot a entrado a nuestro metaverso, recordemos que el bot utiliza el avatar giaa Quixote que es el master de este metaverso, por lo tanto vemos que en nuestro metaverso se pueden conectar varios usuarios que estén dados de alta en nuestro metaverso:



7.11. Configurar el navegador para el Metaverso UC3M

Vamos a configurar el navegador para que los alumnos puedan conectarse fácilmente al Metaverso UC3M.

- Hacemos clic en el navegador, por supuesto no puede ser el navegador de Linden Lab, pues a ellos no les interesa que accedamos a otros metaversos:



- Vemos que tenemos el siguiente cuadro de control:

First Name: saga
Last Name: Bayn
Password: ****************
Log In
Grids
Start Location: My Last Location
Remember password
Quick Grid Select: secondlife

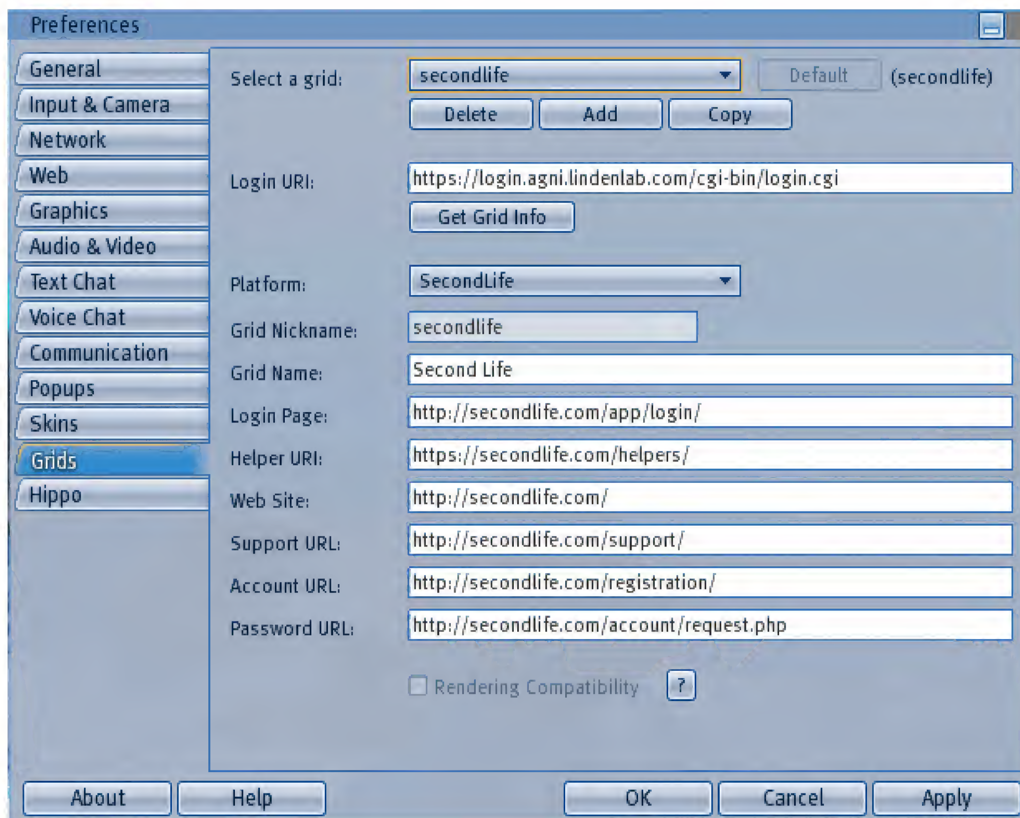
- Vemos los metaversos que tenemos, haciendo clic en la pestaña "**Quick Grid Select**":



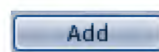
- Hacemos clic en el botón "**Grids**":



- Nos aparece el siguiente cuadro:



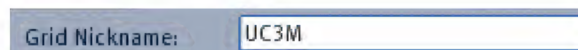
- Hacemos clic en el botón **Add**:



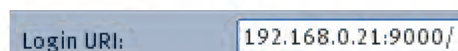
- Damos el nombre al Grid:



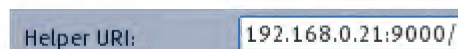
- Introducimos el Grid Nickname:



- Introducimos la dirección IP y el puerto de nuestro metaverso en el campo Login URI:



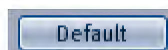
- Introducimos la dirección IP y el puerto de nuestro metaverso en el campo Helper URI:



- Hacemos clic en el botón Apply, para aplicar los cambios:



- Hacemos clic en el botón "**Default**" para fijar por defecto este mundo virtual, pero eso no significa que no puedan conectarse a otros mundos virtuales:



- Podemos ver como el metaverso UC3M se ha agregado a los otros mundos virtuales:



- Hacemos clic en el botón "**OK**":



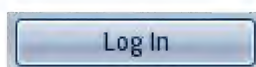
- Vemos que en el listado ya aparece uc3m, y aparece por defecto:



- Introducimos los datos de nuestro avatar, y nos fijamos que nos vamos a conectar al metaverso UC3M pues es el que hemos puesto que se seleccione por defecto, es decir, se seleccionará siempre que abramos el navegador:



- Hacemos clic en el botón conectar, para conectarnos al metaverso de la UC3M:



- Ya estamos conectados al metaverso de la UC3M:



7.12. Crear la grid UC3M

Vamos a crear una grid para tener nuestro propio metaverso al cual se podrán unir otros metaversos. La grid que vamos a crear va a ser la grid de la universidad Carlos III de Madrid.

De esta manera podríamos crear una grid de la universidad UC3M totalmente gratuita que esté formada por distintos metaversos que sean las distintas universidades tanto de España como de todas las partes del mundo.

La gran ventaja es que estarían unidas en un solo metaverso de forma totalmente gratuita todas las universidades, la cual estaría formada por los diferentes profesores y alumnos de las distintas universidades que quieran participar.

Una ventaja de construir esta grid, es que sería propia, con lo cual no se dependería de nadie, y además al restringir los avatares que pueden formar parte de esta grid, es decir, profesores y alumnos de las distintas universidad, protegeríamos a los alumnos, pues sería una grid totalmente segura, pues estaría controlado toda persona que accede a ella y también todos los actos que realizan los avatares.

7.12.1. Configurar OpenSim en modo Grid con MySql

La configuración de MySQL es muy diferente dependiendo de si tiene intención de funcionar OpenSim de manera autónoma (standalone) o en red (grid):

- **OpenSim en modo standalone (autónomo).**

  Es necesario proporcionar un numero de parámetros al fichero OpenSim.ini.

- **OpenSim en modo grid (red).**

  En modo grid cada uno de los servidores UGAIM tiene su propio archivo de configuración que incluye una cadena de conexión MySql y cada región tiene un servidor OpenSim que también contiene una o más cadenas de conexión MySql.

7.12.1.1. Configuración de MySql en modo Grid

Servidores UGAIM.

Cada uno de los servidores UGAIM tiene su propio archivo de configuración archivo que se encuentra en el directorio bin de OpenSim.

La buena noticia es que si este archivo no existe (por ejemplo, la primera vez que el servidor se ejecuta) el operador de consola le pedirá que proporcione los parámetros y por lo general las opciones por defecto funcionan bien.

La mala noticia es que la cadena de conexión de base de datos (database connection string) es el más complicado de introducir y no tiene un valor por defecto.

La otra buena noticia es que estos archivos de configuración se almacenan en formato XML en archivos de texto sin formato para que puedas editarlo con tu texto favorito o un editor XML.

Estos ficheros se llaman UserServer_Config.xml, GridServer_Config.xml, AssetServer.xml, InventoryServer_Config.xml y MessagingServer_Config.xml.

Suponiendo que se permite la configuración de ficheros UGAIM que se genera automáticamente en lugar de crearlos a mano, hay dos parámetros para cada servidor que se refieren a la base de datos.

En primer lugar el plug-in de almacenamiento se debe configurar en **OpenSim.Data.MySQL.dll** el cual es normalmente por defecto.

En segundo lugar viene la cadena de conexión de base de datos (database connection string). Se trata esencialmente de una serie de parámetros comprimido en un objeto punto y coma de valores separados.

Así que una cadena de base de datos de conexión típica sería algo así:

```
data source=localhost; database=opengrid; user id=griduser; password=password; pooling=false; port=3306;
```

Desafortunadamente esta es una cadena larga y tiene que ser de introducida 5 veces (una vez para cada uno de los servidores UGAIM).

Los valores importantes son:

- **Data source.**

  Este es el nombre del host, tal como se define en el fichero My.cnf. Por lo general, "localhost" o la dirección IP del servidor de base de datos. Tenga en cuenta que a MySQL "localhost" y "127.0.0.1" tienen diferentes significados, lo que podría parecer contrario a la intuición. Para MySQL, "localhost" significa utilizar un archivo de socket local, y "127.0.0.1" significa usar TCP.

- **Database.**

  Este es el nombre de la base de datos Grid. Las instrucciones para crear la base de datos utilice solo el mismo nombre escogido y usado allí.

- **User id.**

  El nombre del usuario que utiliza este servidor. Usted puede utilizar el mismo id de usuario para todos los servidores o elegir uno diferente para cada servidor el cual puede facilitar el seguimiento de las sesiones.

- **Password.**

  La contraseña para el ID de usuario.

- **pooling**

  Puede ser verdadera o falsa. Define si los usuarios comparten un grupo de conexiones o conectar / desconectar individualmente. Su valor predeterminado es cierto lo cual pueden tener ventajas de rendimiento cuando las conexiones son con frecuencia se hacen y se rompen, pero no es así como OpenSim está pensado para trabajar, así que por lo general se establece en false.

- **Port** (puerto).

  El valor por defecto es 3306, que sólo necesita ser cambiado si el defecto no está seleccionado en el archivo de configuración de MySQL.

La creación de la base de datos.

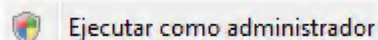
Los servidores opensimulator UGAIM crearán las tablas en la base de datos donde ellas no existen, pero no va a crear la propia base de datos. Por lo tanto, es necesario crearla manualmente. Esto se puede lograr ya sea usando la interfaz gráfica de usuario administrador de MySQL, o manualmente desde el símbolo del sistema:

- Vamos a crear la base de datos OpenSim mediante el programa WampServer:

  - Si trabajamos en Vista seleccionamos con el botón derecho del ratón el acceso directo de WampServer:



  - Hacemos clic en **Ejecutar como administrador**:

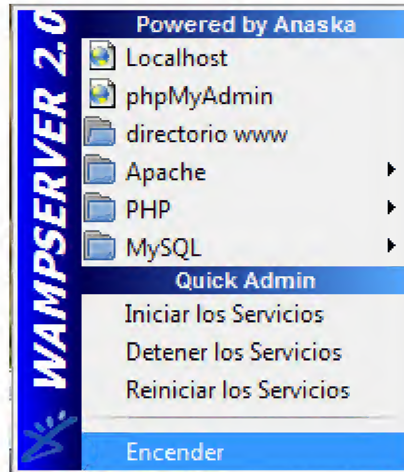


Por línea de comando sería **$ mysql -u root -p -h localhost**.

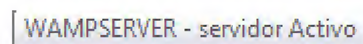
- Nos sale el siguiente icono:



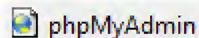
- Hacemos clic en el icono, y en el menú que nos sale damos a **encender**:



- Al pasar el ratón por encima del icono de WampServer, nos saldrá un mensaje que nos indicara: "WampServer – Servidor Activo":



- Seleccionamos la opción phpMyAdmin, para administrar la base de datos MySql:



- Hacemos clic en la pestaña SQL:



- Vemos el siguiente cuadro de dialogo que sirve para crear una base de datos:



- Introducimos el nombre de OpenSim:

opengrid

- Hacemos clic en el botón Crear:

Crear

- Nos sale el siguiente mensaje por pantalla indicándonos que la base de datos se ha creado:

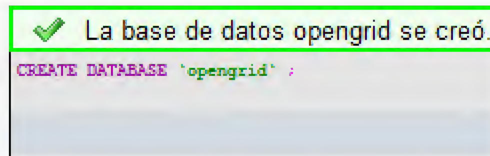


Figura 7.12. Crear base de datos en OpenSim.

Por línea de comando sería **mysql> CREATE DATABASE opengrid;**

- Vemos que ya tenemos nuestra base de datos creada:

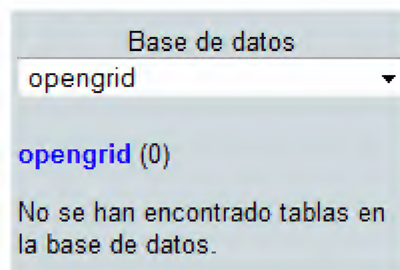


Figura 193. Base de datos para OpenSim.

Por supuesto, será necesario proporcionar la contraseña de root. El nombre de la base de datos y el nombre de host coincidan con la cadena de conexión de la base de datos (database connect string) dado a los servidores de UGAIM. También es posible usar el usuario root para conectarse a MySQL desde los servidores de UGAIM o puede crear usuarios específicos para este fin:

mysql> CREATE USER griduser IDENTIFIED BY 'password';

mysql> GRANT ALL PRIVILEGES ON opengrid.* TO griduser;

El comando GRANT permite al usuario acceso completo a la base de datos, pero puede ser necesario que se repita después de que las tablas sean creadas.

- Nuestra **cadena de conexión a la base de datos** es la siguiente:

  Data source=localhost; database=opengrid; user id=root; Password = ; pooling=false; port=3306;

7.12.2. Ejecución de OpenSim en Modo Grid

Los Grid servidores necesitan crearse y la configuración de la región simulada necesita ser cambiada para usar instancias de construcciones independientes (stadanalone) en modo únicas.

7.12.2.1. Configuración de los servicios Grid.

R.O.B.U.S.T. tiene conectores para la funcionalidad de activo (A), grid (G) y el inventario (I). Los servidores aún se requieren funcionalidad para el usuario (U) y mensajería (M).

En los siguientes pasos te mostramos como crearlo y ejecutarlo. Vamos a suponer que el directorio actual es bin/ dentro de la distribución OpenSim.

1. Configura el UserServer.

Vamos a configurar el UserServer.

- Arrancamos el OpenSim.Grid.UserServer.exe:



- Te preguntaran un número de cuestiones de configuración. Si los servidores de la red están todos en una sola máquina, la mayoría de los valores por defecto son seguros de usar.

- La configuración se guarda en el archivo UserServer_Config.xml que se puede cambiar posteriormente si es necesario.



  - Nos aparece la siguiente pantalla:

```
OpenSim.Grid.UserServer.exe
00:21:04 - Launching UserServer...
00:21:04 - [STARTUP]: Beginning startup processing
00:21:04 - [STARTUP]: Version: OpenSim 0.6.8 (Release)

USER SERVER: Default Startup Message [Welcome to OGS]:
```

  - Estamos como usuario servidor, y nos pregunta si queremos que el mensaje que se ejecute sea "Bienvenido a OGS" (Las siglas OGS significan OpenSim Grid UserServer), pulsamos la tecla enter para admitir ese mensaje, que es el mensaje por defecto:

```
USER SERVER: Default Startup Message [Welcome to OGS]:
```

  - Nos pregunta si la URL por defecto del Servidor Grid es http://127.0.0.1:8003/:

```
USER SERVER: Default Grid Server URI [http://127.0.0.1:8003/]:
```

  - Introducimos que la URL por defecto del Servidor Gris se nuestra IP interna (por que las pruebas que estamos realizando es en una red LAN, si quisiéramos dar cobertura en internet utilizaríamos la IP externa):

```
USER SERVER: Default Grid Server URI [http://127.0.0.1:8003/]: http://192.168.0.
16:8003/
```

Nota:

Recordemos que para obtener el valor de nuestra IP interna nos introducimos en la consola de MS-DOS, y ponemos el comando **cmd**, y a continuación ponemos **ipconfig**

```
Dirección IPv4. . . . . . . . . . . . . . : 192.168.0.16
```

Si queremos obtener el valor de nuestra IP externa nos vamos a la siguiente dirección web: http://www.cual-es-mi-ip.net/

mi ip
89.141.4.35

- Nos pregunta si la clave que se tiene que enviar al servidor grid es null, hacemos clic en la tecla enter para aceptar:

```
USER SERVER: Key to send to grid server [null]:
```

- Nos pregunta por la clave que esperemos que nos envíe el servidor grid, hacemos clic en la tecla enter para aceptar:

```
USER SERVER: Key to expect from grid server [null]:
```

- Nos pregunta por la dirección URL del inventario del servidor:

```
USER SERVER: Default Inventory Server URI [http://127.0.0.1:8003/]:
```

- Introducimos la dirección IP interna de nuestra máquina porque estamos trabajando en una LAN y vamos a construir una grid de OpenSim para una red LAN (es idéntico en internet, pero en internet tenemos que introducir nuestra IP externa) y el puerto 8003:

```
USER SERVER: Default Inventory Server URI [http://127.0.0.1:8003/]: http://192.1
68.0.16:8003/
```

- Nos pregunta por la URL externa del usuario del servidor para la autentificación de las claves, y damos a la tecla enter:

```
USER SERVER: User Server (this) External URI for authentication keys [http://loc
alhost:8002/]:
```

- Nos pregunta por la ruta al control de archivos de la biblioteca, y damos a la tecla enter:

```
USER SERVER: Path to library control file [.\inventory\Libraries.xml]:
```

- Nos pregunta por la Biblioteca de enlace dinámico (DLL) del provedor de base de datos, y damos a la tecla enter para aceptar que la base de datos con la que vamos a trabajar sea MySql:

```
USER SERVER: DLL for database provider [OpenSim.Data.MySQL.dll]:
```

- Nos piden la cadena de caracteres para conectarnos al base de datos:

```
USER SERVER: Connection String for Database []:
```

- Introducimos la cadena de conexión a la base de datos:

```
USER SERVER: Connection String for Database []: data source=localhost;database=o
pengrid;user id=root;password= ;pooling=false;port=3306;
```

- Nos pregunta si el puerto http de escucha va a ser el 8002, y damos a la tecla enter para aceptar:

```
USER SERVER: Http Listener port [8002]:
```

- Nos pregunta si queremos usar SSL (Secure Sockets Layer –Protocolo de capa de conexión segura-), damos a la tecla enter para confirmar que no queremos utilizarlo:

```
USER SERVER: Use SSL? true/false [False]:
```

- Nos pregunta si es buena la región X en la coordenada 1000, y damos a la tecla enter para aceptar:

```
USER SERVER: Known good region X [1000]:
```

- Nos pregunta si es buena la región Y en la coordenada 1000, y damos a la tecla enter para aceptar:

```
USER SERVER: Known good region Y [1000]:
```

- Nos pregunta si habilitamos el soporte para conectar las LLSD , que se usa normalmente por librerías basadas en clientes y robot (bot), damos a la tecla enter para indicar que si la queremos habilitar:

```
USER SERVER: Enable LLSD login support [Currently used by libsl based clients/bo
ts]? true/false [True]:
```

- Nos pregunta si queremos activar el soporte para conectar Hypergrid, y hacemos clic en aceptar:

```
USER SERVER: Enable Hypergrid login support [Currently used by GridSurfer-proxie
d clients]? true/false [True]:
```

- Nos pregunta por el mínimo nivel de usuarios que deberíamos tener conectados, damos a la tecla enter para aceptar que cero:

```
USER SERVER: Minimum Level a user should have to login [0 default] [0]:
```

- Nos preguntas si queremos habilitar el acceso remoto por consola del nombre de usuario, damos a la tecla enter para indicar que permanezca deshabilitado:

```
USER SERVER: Remote console access user name [Default: disabled] []:
```

os pregunta si queremos habilitar el acceso remoto por consola de la clave, damos a la tecla enter para indicar que permanezca desabilitado:

```
USER SERVER: Remote console access password [Default: disabled] []:
```

- Nos sale la siguiente información:

```
23:18:03 - [STARTUP]: Establishing data connection
23:18:03 - [PLUGINS]: Initializing addin manager
23:18:06 - [PLUGINS]: Loading extension point /OpenSim/UserData
23:18:06 - [PLUGINS]: Trying plugin /OpenSim/UserData/SQLiteUserData
23:18:06 - [PLUGINS]: Trying plugin /OpenSim/UserData/MySQLUserData
23:18:06 - [PLUGINS]: Plugin Loaded: OpenSim.Data.MySQL
23:18:06 - [PLUGINS]: Trying plugin /OpenSim/UserData/MSSQLUserData
23:18:06 - [MYSQL]: Connection established
23:18:06 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:06 - Creating 10 DB connections...
23:18:06 - Connecting to DB... [0]
23:18:07 - [MYSQL]: Connection established
23:18:07 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:07 - Connecting to DB... [1]
23:18:07 - [MYSQL]: Connection established
23:18:07 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:07 - Connecting to DB... [2]
23:18:08 - Connecting to DB... [4]
23:18:09 - [MYSQL]: Connection established
23:18:09 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:09 - Connecting to DB... [5]
23:18:09 - [MYSQL]: Connection established
23:18:09 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:09 - Connecting to DB... [6]
23:18:10 - [MYSQL]: Connection established
23:18:10 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:10 - Connecting to DB... [7]
23:18:10 - [MYSQL]: Connection established
23:18:10 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:10 - Connecting to DB... [8]
23:18:11 - [MYSQL]: Connection established
23:18:11 - [REGION DB]: Connection wait timeout 28860 seconds
23:18:11 - Connecting to DB... [9]
23:18:11 - [MIGRATIONS]: Creating UserStore at version 1
23:18:11 - [MIGRATIONS]: Updating UserStore to version 2
23:18:11 - [MIGRATIONS]: Updating UserStore to version 3
23:18:11 - [MIGRATIONS]: Updating UserStore to version 4
23:18:11 - [MIGRATIONS]: Updating UserStore to version 5
23:18:12 - [MIGRATIONS]: Updating UserStore to version 6
23:18:12 - [MIGRATIONS]: Updating UserStore to version 7
23:18:12 - [MIGRATIONS]: Updating UserStore to version 8
23:18:12 - [GRID INFO SERVICE]: Grid info service initialized with 5 keys
23:18:12 - [LIBRARY INVENTORY]: Loading library control file .\inventory\Librari
es.xml
23:18:12 - [LIBRARY INVENTORY]: Loading library control file .\inventory\Librari
es.xml
23:18:12 - [GRID]: Login Level set to 0
23:18:12 - [STARTUP]: Starting HTTP process
23:18:12 - [BASE HTTP SERVER]: Starting up HTTP Server
23:18:12 - [BASE HTTP SERVER]: Spawned main thread OK
Console log level is ALL
23:18:12 - [STARTUP]: Startup took 7m 30s
Enter help for a list of commands

User#
```

- El servidor de usuario entonces se carga, y muestra esto en la ventana de comandos:

```
User#
```

- Vemos que se han creado las siguientes tablas:

Estructura | SQL | Buscar | Generar una consulta | Exportar | Importar | Operaciones | Privilegios | Eliminar

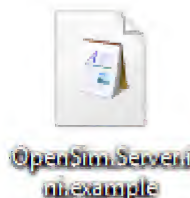
Tabla	Acción	Registros [1]	Tipo	Cotejamiento	Tamaño	Residuo a depurar
agents		0	InnoDB	utf8_general_ci	48.0 KB	-
avatarappearance		0	InnoDB	utf8_general_ci	16.0 KB	-
avatarattachments		0	InnoDB	latin1_swedish_ci	16.0 KB	-
migrations		2	MyISAM	latin1_swedish_ci	1.0 KB	-
userfriends		0	InnoDB	utf8_general_ci	16.0 KB	-
users		0	InnoDB	utf8_general_ci	32.0 KB	-
6 tabla(s)	Número de filas	2	MyISAM	latin1_swedish_ci	129.0 KB	0 Bytes

Marcar todos/as / Desmarcar todos   Para los elementos que están marcados:

2. Configuración de R.O.B.U.S.T.

Vamos a configurar ROBUST.

- Vamos a la carpeta bin, y seleccionamos el archive **OpenSim.Server.ini.example**:



- Abrimos el **OpenSim.Server.ini.example** y vemos que la siguiente cadenas de conexión de la base de datos:

```
StorageProvider = "OpenSim.Data.MySQL.dll"
ConnectionString = "Data Source=localhost;Database=grid;User
ID=grid;Password=grid;"
```

Aparecen en las secciones de AssetService, InventoryService, AuthenticationService, UserAccountService y GridService, hay que cambiarlo en todas la secciones.

- Cambiamos la cadena de conexión de la base de datos, por la que tenemos configurada en nuestra base de datos:

```
ConnectionString = "Data Source=localhost;Database=opengrid;User
ID=root;Password= ;"
```

- Buscamos la siguiente línea:

```
UserServerURI = "http://127.0.0.1:8002"
```

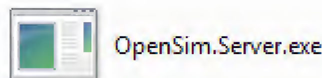
- Introducimos la dirección IP interna (trabajamos en una red LAN):

```
UserServerURI = "http://192.168.0.18:8002"
```

- Cambia el fichero OpenSim.Server.ini.example to **OpenSim.Server.ini**:



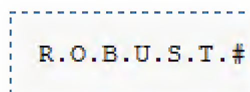
- Ejecuta OpenSim.Server.exe:



- Vemos la siguiente pantalla:

```
00:25:53 - [SERVER]: GridServiceConnector loaded successfully
R.O.B.U.S.T.#
```

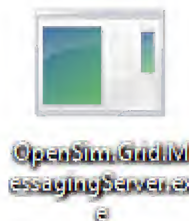
- El servidor se pondrá en marcha en activos, inventario y los componentes de la grid y mostrar el sistema robusto:



3. Configurar el servidor de mensajería.

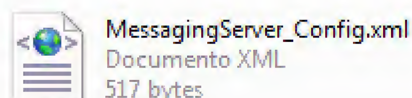
Vamos a configurar el servidor de mensajería.

- Ejecuta OpenSim.Grid.MessagingServer.exe que esta en la carpeta bin:



- Responda a las preguntas, si todos los servidores se ejecutan en la misma máquina, es seguro, presione ENTER para todas las preguntas.

- La configuración se almacena en **MessagingServer_Config.xml**.



- Nos abre la siguiente ventana:

```
C:\opensim-0.6.8-binaries\bin\OpenSim.Grid.MessagingServer.exe
01:34:26 - [SERVER]: Launching MessagingServer...
01:34:26 - [STARTUP]: Beginning startup processing
01:34:26 - [STARTUP]: Version: OpenSim 0.6.8 (Release)

MESSAGING SERVER: Default User Server URI [http://127.0.0.1:8002/]:
```

- Nos pide que introduzcamos la dirección url del servidor por defecto:

```
MESSAGING SERVER: Default User Server URI [http://127.0.0.1:8002/]:
```

- Introducimos la dirección IP interna (porque estamos trabajando en una LAN, si trabajáramos en internet introduciríamos la dirección IP externa):

```
MESSAGING SERVER: Default User Server URI [http://127.0.0.1:8002/]: http://192.1
68.0.16:8003/
```

- Nos pregunta si la clave a enviar al servidor de usuario es null, y damos a la tecla enter para confirmar:

```
MESSAGING SERVER: Key to send to user server [null]:
```

- Nos pregunta si la clave que esperamos que nos envíe el servidor de usuario es null, damos a la tecla enter para confirmar:

```
MESSAGING SERVER: Key to expect from user server [null]:
```

- Nos pregunta si la URL por defecto del Servidor Grid es http://127.0.0.1:8003/:

```
MESSAGING SERVER: Default Grid Server URI [http://127.0.0.1:8003/]:
```

- Introducimos que la URL por defecto del Servidor Gris se nuestra IP interna (por que las pruebas que estamos realizando es en una red LAN, si quisiéramos dar cobertura en internet utilizaríamos la IP externa):

```
MESSAGING SERVER: Default Grid Server URI [http://127.0.0.1:8003/]: http://192.1
68.0.16:8003/
```

- Nos pregunta si la clave que se tiene que enviar al servidor grid es null, hacemos clic en la tecla enter para aceptar

```
MESSAGING SERVER: Key to send to grid server [null]:
```

- Nos pregunta por la clave que esperemos que nos envíe el servidor grid, hacemos clic en la tecla enter para aceptar:

```
MESSAGING SERVER: Key to expect from grid server [null]:
```

- Nos piden la cadena de caracteres para conectarnos al base de datos:

```
MESSAGING SERVER: Connection String for Database []:
```

- Introducimos la cadena de conexión a la base de datos:

```
MESSAGING SERVER: Connection String for Database []: data source=localhost;datab
ase=opengrid;user id=root;password= ;pooling=false;port=3306;
```

- Nos pregunta por la Biblioteca de enlace dinámico (DLL) del provedor de base de datos, y damos a la tecla enter para aceptar que la base de datos con la que vamos a trabajar sea MySql:

```
MESSAGING SERVER: DLL for database provider [OpenSim.Data.MySQL.dll]:
```

- Nos pregunta por la DDL para el provedor comms, damos a la tecla enter para confirmar que utilizaremoes el OpenSim.Region.Communications.OGS1.dll:

```
MESSAGING SERVER: DLL for comms provider [OpenSim.Region.Communications.OGS1.dll
]:
```

- Nos pregunta si el puerto http de escucha va a ser el 8006, y damos a la tecla enter para aceptar:

```
MESSAGING SERVER: Http Listener port [8006]:
```

- Nos pregunta si queremos usar SSL (Secure Sockets Layer –Protocolo de capa de conexión segura-), damos a la tecla enter para confirmar que no queremos utilizarlo:

```
MESSAGING SERVER: Use SSL? true/false [False]:
```

- Nos pregunta por nuestra dirección IP publica:

```
MESSAGING SERVER: My Published IP Address [127.0.0.1]:
```

- Introducimos nuestra IP interna (si trabajas en una red LAN):

```
MESSAGING SERVER: My Published IP Address [127.0.0.1]: 192.168.0.16
```

- Nos preguntas si queremos habilitar el acceso remoto por consola del nombre de usuario, damos a la tecla enter para indicar que permanezca deshabilitado:

```
MESSAGING SERVER: Remote console access user name [Default: disabled] []:
```

- Nos pregunta si queremos habilitar el acceso remoto por consola de la clave, damos a la tecla enter para indicar que permanezca desabilitado:

```
MESSAGING SERVER: Remote console access password [Default: disabled] []:
```

- Nos sale la siguiente información:

```
01:54:08 - [PLUGINS]: Initializing addin manager
01:54:09 - [PLUGINS]: Loading extension point /OpenSim/UserData
01:54:09 - [PLUGINS]: Trying plugin /OpenSim/UserData/SQLiteUserData
01:54:09 - [PLUGINS]: Trying plugin /OpenSim/UserData/MySQLUserData
01:54:09 - [PLUGINS]: Plugin Loaded: OpenSim.Data.MySQL
01:54:09 - [PLUGINS]: Trying plugin /OpenSim/UserData/MSSQLUserData
01:54:10 - Creating 10 DB connections...
01:54:10 - Connecting to DB... [0]
01:54:10 - [MYSQL]: Connection established
01:54:10 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:10 - Connecting to DB... [1]
01:54:11 - [MYSQL]: Connection established
01:54:11 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:11 - Connecting to DB... [2]
01:54:11 - [MYSQL]: Connection established
01:54:11 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:11 - Connecting to DB... [3]
01:54:12 - [MYSQL]: Connection established
01:54:12 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:12 - Connecting to DB... [4]
01:54:12 - [MYSQL]: Connection established
01:54:13 - [MYSQL]: Connection established
01:54:13 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:13 - Connecting to DB... [7]
01:54:14 - [MYSQL]: Connection established
01:54:14 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:14 - Connecting to DB... [8]
01:54:14 - [MYSQL]: Connection established
01:54:14 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:14 - Connecting to DB... [9]
01:54:15 - [MYSQL]: Connection established
01:54:15 - [REGION DB]: Connection wait timeout 28860 seconds
01:54:15 - [MIGRATIONS]: UserStore up to date, no migrations to apply
01:54:16 - [SERVER]: Unable to connect to server http://192.168.0.16:8003/ for r
egistration. User service not running?  Exception No es posible conectar con el
servidor remoto    en System.Net.HttpWebRequest.GetRequestStream()
   en Nwc.XmlRpc.XmlRpcRequest.Send(String url, Int32 timeout)
   en Nwc.XmlRpc.XmlRpcRequest.Invoke(String url)
   en OpenSim.Grid.MessagingServer.Modules.InterMessageUserServerModule.register
WithUserServer()
01:54:16 - [STARTUP]: Unable to connect to User Server
01:54:16 - [SERVER]: Messageserver 0.5 - Startup complete
Console log level is ALL
01:54:16 - [STARTUP]: Startup took 19m 50s
Enter help for a list of commands

Messaging#
```

- El servidor se iniciará y le presentará con su sistema:

```
Messaging#
```

7.12.2.2. Resumen de los comandos para ejecutar los ejecutables servicios de grid

Vamos a ejecutar en tres consolas distintas OpenSim.Grid.UserServer, OpenSim.Server.exe y OpenSim.Grid.MessagingServer.exe.

- Abrimos la consola de commandos y entramos en la carpeta opensim-0.6.8-binaries\bin:

```
C:\opensim-0.6.8-binaries\bin>
```

- Escribimos OpenSim.Grid.UserServer.exe:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.Grid.UserServer.exe
```

- No ha dado ningún error (no letras rojas) y esta listo para introducir comandos:

```
02:26:07 - [STARTUP]: Startup took 0m 6s
Enter help for a list of commands

User#
```

- Abrimos otra consola de commandos y entramos en la carpeta opensim-0.6.8-binaries\bin:

```
C:\opensim-0.6.8-binaries\bin>
```

- Escribimos OpenSim.Server.exe:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.Server.exe
```

- No ha dado ningún error (no letras rojas) y esta listo para introducir comandos:

```
02:52:56 - [BASE HTTP SERVER]: Starting up HTTP Server
02:52:56 - [BASE HTTP SERVER]: Spawned main thread OK
02:52:56 - [SERVER]: Loading AssetServiceConnector
02:52:56 - [MYSQL]: Connection established
02:52:56 - [REGION DB]: Connection wait timeout 28860 seconds
02:52:56 - [MIGRATIONS]: Upgrading AssetStore to latest revision.
02:52:56 - [MIGRATIONS]: NOTE: this may take a while, don't interupt this proces
s!
02:52:57 - [MIGRATIONS]: Updating AssetStore to version 3
02:52:57 - [MIGRATIONS]: Updating AssetStore to version 4
02:52:57 - [MIGRATIONS]: Updating AssetStore to version 5
02:52:57 - [MIGRATIONS]: Updating AssetStore to version 6
02:52:57 - [ASSET]: Loading default asset set from assets/AssetSets.xml
02:53:01 - [ASSET CONNECTOR]: Local asset service enabled
02:53:01 - [SERVER]: AssetServiceConnector loaded successfully
02:53:01 - [SERVER]: Loading InventoryServiceInConnector
02:53:01 - [MYSQL]: Connection established
02:53:01 - [REGION DB]: Connection wait timeout 28860 seconds
02:53:01 - [MIGRATIONS]: Upgrading InventoryStore to latest revision.
02:53:01 - [MIGRATIONS]: NOTE: this may take a while, don't interupt this proces
s!
02:53:02 - [MIGRATIONS]: Updating InventoryStore to version 4
02:53:02 - [INVENTORY SERVICE]: Initialized.
02:53:02 - [INVENTORY HANDLER]: handlers initialized
02:53:02 - [SERVER]: InventoryServiceInConnector loaded successfully
02:53:02 - [SERVER]: Loading FreeswitchServerConnector
02:53:02 - [SERVER]: FreeswitchServerConnector loaded successfully
02:53:02 - [SERVER]: Loading GridServiceConnector
02:53:02 - [MIGRATIONS]: Upgrading GridStore to latest revision.
02:53:02 - [MIGRATIONS]: NOTE: this may take a while, don't interupt this proces
s!
02:53:02 - [MIGRATIONS]: Updating GridStore to version 4
02:53:02 - [GRID SERVICE]: Starting...
02:53:02 - [SERVER]: GridServiceConnector loaded successfully
R.O.B.U.S.T.#
```

- Abrimos otra consola de commandos y entramos en la carpeta opensim-0.6.8-binaries\bin:

```
C:\opensim-0.6.8-binaries\bin>
```

- Escribimos OpenSim.Grid.MessagingServer.exe:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.Grid.MessagingServer.exe
```

- No ha dado ningún error (no letras rojas) y esta listo para introducir comandos:

```
02:59:20 - [STARTUP]: Unable to connect to User Server
02:59:20 - [SERVER]: Messageserver 0.5 - Startup complete
Console log level is ALL
02:59:21 - [STARTUP]: Startup took 0m 6s
Enter help for a list of commands

Messaging#
```

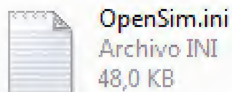
7.12.2.3. Configurar regiones

Cuando se ejecuta la primera vez, una nueva instalación de OpenSim le hará una serie de preguntas para establece una región.

Configurando y arrancando el simulador de la región.

Vamos a configurar y arrancar el simulador de la región.

- Abre el archivo OpenSim.ini que está en la carpeta bin:



- Encuentra la sección [Architecture]:

```
[Architecture]
    ; Choose exactly one and only one of the architectures below.

    ;Include-Standalone    = "config-include/Standalone.ini"
    ;Include-HGStandalone = "config-include/StandaloneHypergrid.ini"
    ;Include-Grid          = "config-include/Grid.ini"
    ;Include-HGGrid        = "config-include/GridHypergrid.ini"
```

- Busca la línea que dice:

```
Include-Standalone    = "config-include/Standalone.ini"
```

- Comentamos la línea:

```
;Include-Standalone    = "config-include/Standalone.ini"
```

- Busca la línea que dice:

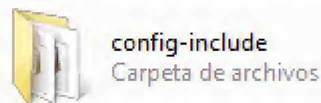
```
;Include-Grid          = "config-include/Grid.ini"
```

- Descomenta la línea:

```
Include-Grid          = "config-include/Grid.ini
```

- Guarda OpenSim.ini.

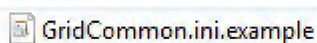
- Ve dentro del directorio config-include:



- Copie los siguientes ficheros ini.example a .ini:

```
GridCommon.ini.example to GridCommon.ini
CenomeCache.ini.example to CenomeCache.ini
FlotsamCache.ini.example to FlotsamCache.ini
```

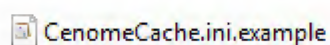
- Hacemos una copia de GridCommon.ini.example:



- La llamamos a la copia GridCommon.ini:



- Hacemos una copia de CenomeCache.ini.example:



- La llamamos a la copia CenomeCache.ini:



- Hacemos una copia de CenomeCache.ini.example:

FlotsamCache.ini.example

- La llamamos a la copia CenomeCache.ini:

FlotsamCache.ini

- Abierto todos estos archivos (Grid.ini, GridCommon.ini, CenomeCache.ini y FlotsamCache.ini) y editar a sus necesidades.

- Abrimos el fichero Grid.ini:

Grid.ini

- Vemos el siguiente mensaje de texto:

```
;;
;; Please don't change this file.
;; All optional settings are in GridCommon.ini.example,
;; which you can copy and change.
;;
```

Nos indican que por favor no cambiemos este fichero. Todas las configuraciones opcionales están en GridCommon.ini.example, el cual tú puedes copiar y cambiar.

- Abrimos el fichero GridCommon.ini:

GridCommon.ini

- Vamos a la sección [AssetService]:

```
[AssetService]
```

- Nos encontramos con la siguiente línea de código:

```
;
; change this to your grid-wide asset server
;
AssetServerURI = "http://myassetserver.com:8003"
```

- La cambiamos por la siguiente línea de código:

```
AssetServerURI = "http://192.168.0.16:8003"
```

- Vamos a la sección [InventoryService]:

```
[InventoryService]
```

- Nos encontramos con la siguiente línea de código:

```
;
; change this to your grid-wide inventory server
;
InventoryServerURI = "http://myinventoryserver.com:8003"
```

- La cambiamos por la siguiente línea de código:

```
InventoryServerURI = "http://192.168.0.16:8003"
```

- Vamos a la sección [GridService]:

```
[GridService]
```

- Nos encontramos con la siguiente línea de código:

```
;
; change this to your grid-wide grid server
;
GridServerURI = "http://mygridserver.com:8003"
```

- La cambiamos por la siguiente línea de código:

```
GridServerURI = "http://192.168.0.16:8003"
```

- Abrimos el fichero CenomeCache.ini:

CenomeCache.ini

- Vemos que nos muestra la siguiente información:

```
[AssetCache]
    ;;
    ;; Options for CenmoeAssetCache
    ;;

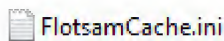
    ; 256 MB (default: 134217728)
    MaxSize =  268435456

    ; How many assets it is possible to store cache (default: 4096)
    MaxCount = 16384

    ; Expiration time - 1 hour (default: 30 minutes)
    ExpirationTime = 60
```

- No cambiamos nada de CenomeCache.ini.

- Abrimos el fichero FlotsamCache.ini:



- No cambiamos nada de FlotasamCache.ini.

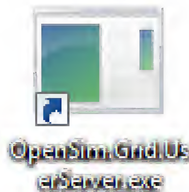
- Ahora debería estar listo para arrancar el Sim. El modo en el que se inicia en OpenSim.exe es normalmente controlada por la configuración de su opensim.ini. El valor predeterminado de OpenSim es en modo standalone (independiente) si en la configuración no se ha especificado o el fichero no se encuentra.

```
C:\opensim-0.6.8-binaries\bin>opensim
```

- Si lo desea, puede obligar a OpenSim para iniciar en modo grid en la línea de comandos de la siguiente manera:

```
OpenSim.exe -gridmode=true
```

- Ve a la consola UserServer (OpenSim.Grid.UserServer):



- Escribe “**create user**” para crear un nuevo avatar:

```
User# create user
```

- Nos pide el nombre:

```
First name [Default]:
```

- Introducimos el nombre del avatar: giaa

```
First name [Default]: giaa
```

- Introducimos el apellido del avatar: Quixote

```
Last name [User]: Quixote
```

- Nos pide la clave del avatar:

```
Password
```

- Introducimos la clave del avatar:

```
Password
```

- Nos pide la X e Y de la Sim que debería ser su localización de la casa (home) del avatar que creamos:

```
Start Region X [1000]:
Start Region Y [1000]:
```

Figura 7.14. Localización X e Y de una sim

Usamos la localización 1000 y 1000, o en la localización donde tu dijiste que se creara tu sim, cuando lo creastes en modo standalone (independiente).

- Nos pide el email:

```
Email []:
```

- Nos piden el e-mail:

```
Email []: e100048464@gmail.com
```

- Nos da la siguiente información:

```
16:06:37 - [MySQLManager]: Fetching profile for 906a1eb3-8916-4cec-ab10-d294b779
aa6f
16:06:37 - [USERSTORAGE]: using m_commsManager.InterServiceInventoryService to c
reate user's inventory
```

- En la consola de cualquiera de estos servidores, usted debería ser capaz de escribir 'help' para obtener una lista de comandos.

```
help
```

- Ahora debería ser capaz de conectarse a la red de nuevo con su cliente de Second Life:



- Usted necesita decirle a su cliente de Second Life para apuntar a la UserServer lugar de hacerlo directamente en el simulador.

- Abrimos el cliente de Second Life con el botón derecho del ratón y introducimos – loginuri http://192.168.0.16:8002:

Destino: anguage es -loginuri http://192.168.0.16:8002

Figura 7.15. Puerto del UserServer

8002 es el Puerto por defecto de UserServer.

Nota: Si se usando Windows vista, recuerda arrancar el los servidores como administrador, para arrancar como administrador damos al botón derecho del ratón y seleccionamos 'arrancar como administrador'. Si no arrancas como administrador te saldrá un aviso de error en la consola como "Error – Acceso denegado".

7.13. Pruebas de OsUC3MGrid

7.13.1. Prueba de acceso a OsUC3MGrid

Vamos a hacer una prueba de entrar en la isla del grid OsUC3MGrid con el avatar giaa Quixote.

- Abrimos una consola y escribimos OpenSim.Grid.UserServer.exe:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.Grid.UserServer.exe
```

- Abrimos otra consola y escribimos OpenSim.Server.exe:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.Server.exe
```

- Abrimos otra consola y escribimos OpenSim.Grid.MessagingServer.exe:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.Grid.MessagingServer.exe
```

- Abrimos otra consola y escribimos OpenSim.exe –gridmode=true:

```
C:\opensim-0.6.8-binaries\bin>OpenSim.exe -gridmode=true
```

- Ya hemos conectado OpenSim en modo grid:

```
C:\Windows\system32\cmd.exe - OpenSim.exe -gridmode=true
23:32:11 - [NEIGHBOUR CONNECTOR]: HelloNeighbour from UC3M, to 1100611139659776.
 Count = 1
23:32:11 - [NEIGHBOUR CONNECTOR]: region handle 1099511628032256 not found
23:32:11 - [NEIGHBOUR CONNECTOR]: region handle 1100611139659520 not found
23:32:11 - [NEIGHBOUR CONNECTOR]: region handle 1100611139659776 not found
23:32:11 - [NEIGHBOUR CONNECTOR]: HelloNeighbour from UC3M, to 1100611139660032.
 Count = 1
23:32:11 - [NEIGHBOUR CONNECTOR]: region handle 1100611139660032 not found
23:32:11 - [GRID CONNECTOR]: GetRegionByPosition 00000000-0000-0000-0000-0000000
00000, 256000-255744 received invalid response
23:32:11 - [INTERGRID]: Failed to inform neighbour 1000-999 that I'm here.
23:32:11 - [GRID CONNECTOR]: GetRegionByPosition 00000000-0000-0000-0000-0000000
00000, 256256-255744 received invalid response
23:32:11 - [GRID CONNECTOR]: GetRegionByPosition 00000000-0000-0000-0000-0000000
00000, 256000-256256 received invalid response
23:32:11 - [INTERGRID]: Failed to inform neighbour 1001-999 that I'm here.
23:32:11 - [GRID CONNECTOR]: GetRegionByPosition 00000000-0000-0000-0000-0000000
00000, 256256-256000 received invalid response
23:32:11 - [INTERGRID]: Failed to inform neighbour 1000-1001 that I'm here.
23:32:11 - [INTERGRID]: Failed to inform neighbour 1001-1000 that I'm here.
23:32:12 - [GRID CONNECTOR]: GetRegionByPosition 00000000-0000-0000-0000-0000000
00000, 256256-256256 received invalid response
23:32:12 - [INTERGRID]: Failed to inform neighbour 1001-1001 that I'm here.
23:32:12 - [REGION]: Enabling logins for UC3M
Region (UC3M) #
```

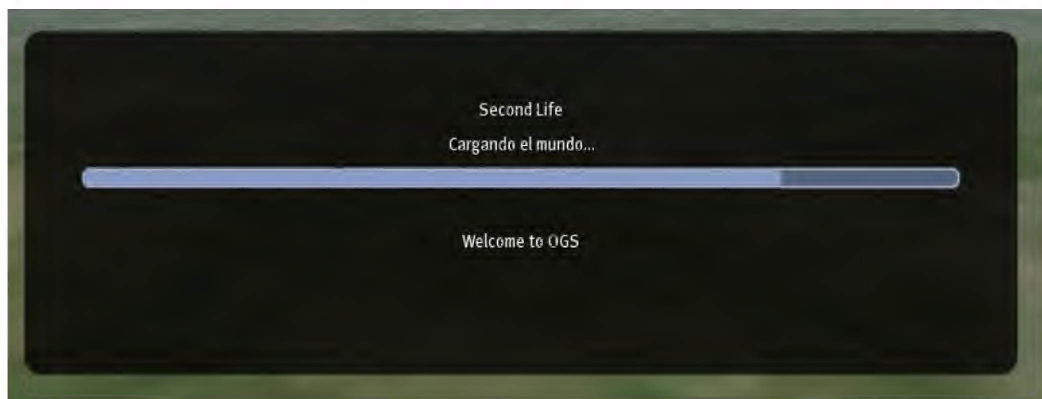
- Abrimos el navegador se Second Life:



- Introducimos la línea de comando http://192.168.0.16:8002:

Destino: anguage es -loginuri http://192.168.0.16:8002

- Vemos como nos dan la bienvenida a OGS, que fue el mensaje de bienvenida que configuramos por defecto:



- Ya tenemos al avatar giaa Quixote en el Sim de OsUC3MGrid:



7.13.2. Prueba de acceso desde otro ordenador de la misma red LAN al OsUC3MGrid

Vamos a entrar en la Sim UC3M del OsUC3MGrid, desde otro ordenador de la misma red LAN que OSUC3MGrid.

- En la consola USER# creo un usuario mediante el comando creater, y le doy la región de su isla:

```
User# create user
First name [Default]: saga
Last name [User]: Bayn
Password1
Start Region X [1000]: 1001
Start Region Y [1000]: 1001
Email []: e100048464@hotmail.com
```

  De esta manera el usuario que cree tendra acceso a la región UC3M.

- Vemos que al crear el usuario nos da la siguiente información:

```
01:17:03 - [MySQLManager]: Fetching profile for 0c28015e-54b0-49e4-8d5a-17e8402c
9add
01:17:03 - [USERSTORAGE]: using m_commsManager.InterServiceInventoryService to c
reate user's inventory
```

- Seleccionamos el cliente Hippo OpenSim:



  Pues cuando tengamos más Sim, nos permite elegir la Sim a la que queremos entrar dándoles las coordenadas o introduciendo el nombre de la Sim.

- Hacemos clic en el botón Grids:



- Seleccionamos la pestaña grid:



- Hacemos clic en el botón Add:



- Introducimos la dirección IP interna del OsUC3MGrid, y el puerto 8002, que es el puerto de los grid:

Login URI: http://192.168.0.16:8002

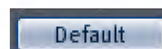
- Introducimos el nombre del Grid al que nos queremos conectar:

Grid Nickname: OsUC3MGrid

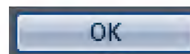
- Hacemos clic en el botón Apply:



- Hacemos clic en el botón Default, para que sea nuestra Grid al que nos conectemos por defecto:



- Damos al botón OK:



- Introducimos los datos del avatar, y la Sim UC3M (también podríamos introducir las coordenadas 1000, 1000), y tenemos seleccionado el grid osuc3mgrid:

First Name: saga | Last Name: Bayn | Password: * | Log In | Grids
Start Location: UC3M | Remember password | Quick Grid Select: osuc3mgrid

- El avatar saga Bayn ha accedido a la región UC3M desde otro ordenador de la misma LAN en la que se encuentra el grid OsUC3Mgrid. Vemos que se encuentra con el avatar giaa Quixote que accede desde el mismo ordenador en que se ejecuta el OsUC3MGrid.



7.13.3. Conectar una Sim al grid OsUC3MGrid

Vamos a conectar una Sim al grid OsUC3MGrid.

- Nos conectamos al ordenador que quiere conectar su sim al grid de OSUC3MGRID.
- Abrimos el archivo OpenSim.ini:



- Vemos que el modo grid esta desactivado:

```
gridmode = false
```

- Activamos el modo grid:

```
gridmode = true
```

- Vemos la opciones del mapa del mundo:

```
; ##
; ## WORLD MAP
; ##

;WorldMapModule = "WorldMap"
;MapImageModule = "MapImageModule"
```

- Activamos las opciones del mapa del mundo:

```
WorldMapModule = "WorldMap"
MapImageModule = "MapImageModule"
```

- Vemos las arquitecturas que tenemos:

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[Architecture]
    ; Choose exactly one and only one of the architectures below.

    Include-Standalone    = "config-include/Standalone.ini"
    ;Include-HGStandalone = "config-include/StandaloneHypergrid.ini"
    ;Include-Grid         = "config-include/Grid.ini"
    ;Include-HGGrid       = "config-include/GridHypergrid.ini"
```

- Vemos que tenemos seleccionada la arquitectura Standalone (individual):

```
;Include-Standalone    = "config-include/Standalone.ini"
```

- Activamos la arquitectura grid quitando las comillas:

```
Include-Grid         = "config-include/Grid.ini"
```

- Abrimos el archivo de la región y le damos una UUID, una localización, una IP interna de nuestro ordenador, un usuario y contraseña:

```
[edu]
RegionUUID = be34f940-79bf-43da-8d9d-085e85ee8f3a
Location = 1001,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 192.168.0.21
MasterAvatarFirstName = sag
MasterAvatarLastName = Bayn
MasterAvatarSandboxPassword = 1
```

- Vemos la sección de network:

```
grid_server_url = "http://127.0.0.1:8003"
grid_send_key = "null"
grid_recv_key = "null"

user_server_url = "http://127.0.0.1:8002"
user_send_key = "null"
user_recv_key = "null"

asset_server_url = "http://127.0.0.1:8003"

inventory_server_url = "http://127.0.0.1:8003"

; The MessagingServer is a companion of the UserServer. It uses
; user_send_key and user_recv_key, too
messaging_server_url = "http://127.0.0.1:8006"
```

- Introducimos la IP interna del grid (no introducimos las claves, porque el OsUC3MGrid no utiliza claves):

```
grid_server_url = "http://192.168.0.16:8003"
grid_send_key = "null"
grid_recv_key = "null"

user_server_url = "http://192.168.0.16:8002"
user_send_key = "null"
user_recv_key = "null"

asset_server_url = "http://192.168.0.16:8003"

inventory_server_url = "http://192.168.0.16:8003"

; The MessagingServer is a companion of the UserServer. It uses
; user_send_key and user_recv_key, too
messaging_server_url = "http://192.168.0.16:8006"
```

- Vamos a la carpeta config-include:



- Copiamos el archivo GridCommon.ini.example:



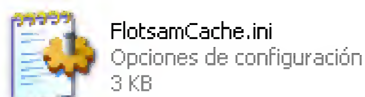
- Cambiamos el nombre de GridCommon.ini.example a GridCommon.ini:



- Copiamos el archivo FlotsamCache.ini.example:



- Cambiamos el nombre de FlotasamCache.ini.example a FlotsamCache.ini:



- Abrimos el fichero GridCommon.ini.

- Vamos la siguiente línea:

```
AssetServerURI = "http://myassetserver.com:8003"
```

- Introducimos la IP interna (porque trabajamos en una red LAN) de la OsUC3MGrid:

```
AssetServerURI = "http://192.168.0.16:8003"
```

- Vamos a la siguiente línea:

```
InventoryServerURI = "http://myinventoryserver.com:8003"
```

- Introducimos la IP interna (porque trabajamos en una red LAN) de la OsUC3MGrid:

```
InventoryServerURI = "http://192.168.0.16:8003"
```

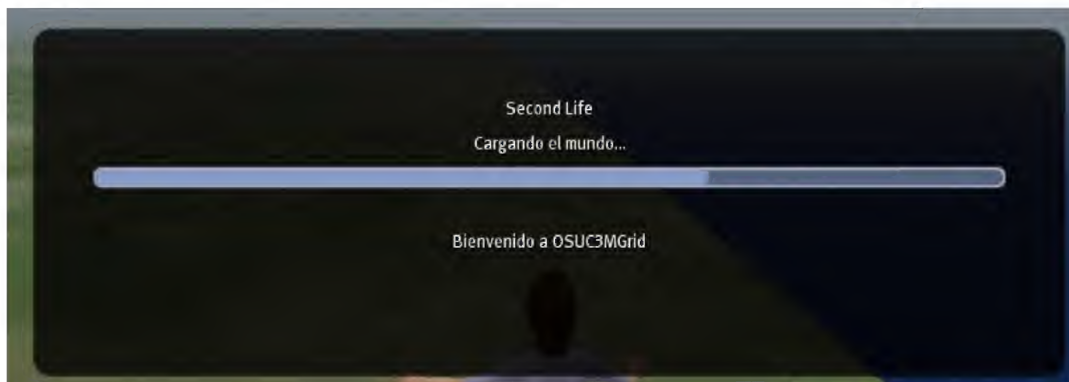
- Vamos a la siguiente línea:

```
GridServerURI = "http://mygridserver.com:8003"
```

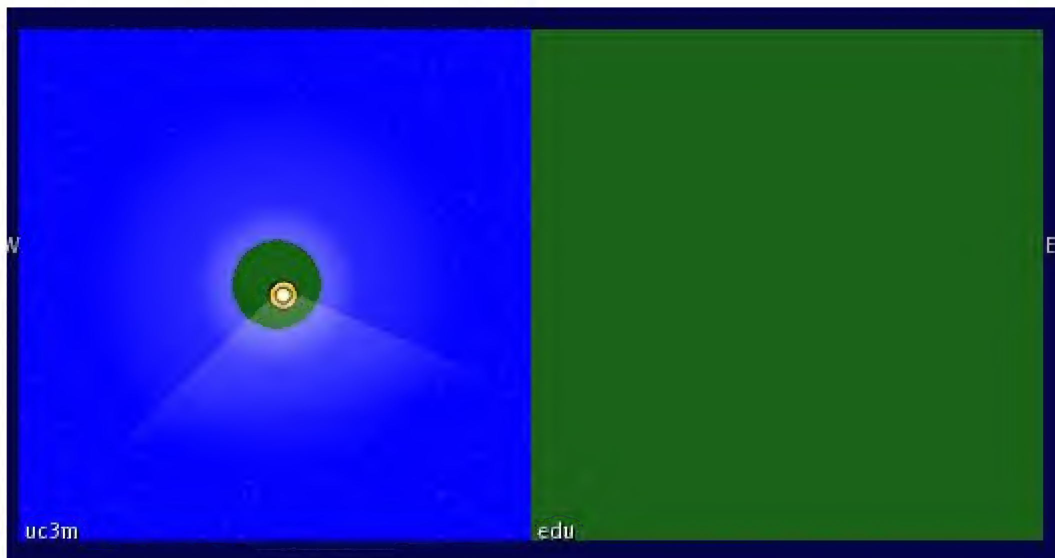
- Introducimos la IP interna (porque trabajamos en una red LAN) de la OsUC3MGrid:

```
GridServerURI = "http://192.168.0.16:8003"
```

- Nos conectamos a OSUC3MGrid mediante el ordenador que está instalado OSUC3MGrid:



- Vemos la región uc3m y la región edu, la región uc3m corre en la maquina en la que se ejecuta el osuc3mgrid, y la región edu corre en otra maquina de la misma red LAN:



- <u>Vemos como desde la región uc3m vemos la región edu</u>:



7.14. El OpenSim Hypergrid

El hypergrid es una extensión de OpenSim que le permite enlazar su OpenSim a opensims otros en el Internet, y que soporta agentes de transferencias sin fisuras entre aquellos opensims.

Se puede utilizar tanto en modo independiente (standAlone) y en modo de grid. El hypergrid apoya eficazmente la aparición de una web de los mundos virtuales.

La idea básica para la hypergrid es que los administradores de una región/grid pueden colocar hipervínculos en su mapa parar hypergrided regiones a cargo de los demás.

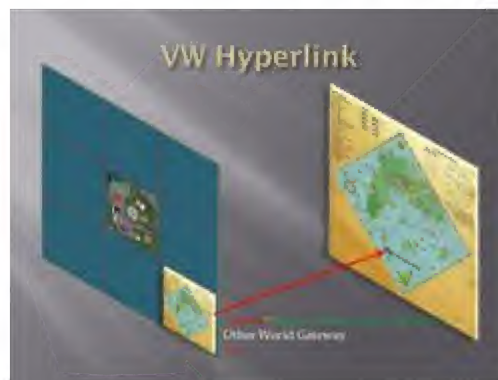
Una vez que los enlaces se establezcan, los usuarios interactúan con las regiones exactamente de la misma manera en que interactúan con las regiones locales. En concreto, los usuarios pueden optar por teleport allí. Una vez que el usuario accede a la región tras el hipervínculo, el esta automáticamente interactuando con un mundo virtual diferente sin tener que salir del sistema desde el mundo de dónde venía, y mientras mantiene el acceso a su inventario.

El hypergrid se incluye en la distribución estándar de OpenSim.

Hipervínculos mundo virtual

Todos estamos familiarizados con enlaces de hipertexto en la Web. Pero, ¿qué es un hipervínculo mundo virtual?

En el modelo hypergrid, consideramos que el mapa 2D del mundo virtual como el equivalente de una página web. Como tal, un enlace VW es simplemente una región en el mapa.



El modelo por defecto de los mundos virtuales basados en OpenSim ya apoya este concepto de hipervínculo, hasta cierto punto. Cuando te teletransportas de una región a otra a través del mapa, es probable que va a migrar a un servidor OpenSim diferentes. Esta migración es un denominado "agente de transferencia" que también existe, en forma rudimentaria, en la web cuando se siguen los enlaces de hipertexto.

En primer lugar, permite a los casos individuales OpenSim para añadir "vecinos" a su mapa local, pasando el control del mapa desde el servidor de la red a los distintos casos OpenSim.

Como administrador OpenSim, usted es libre de definir lo que desee ver en su mapa de OpenSim.

En segundo lugar, permite la transmisión de agentes pertenecientes a usuarios de extranjero, es decir, los usuarios que estén registrados en otros lugares.

Escenarios de uso

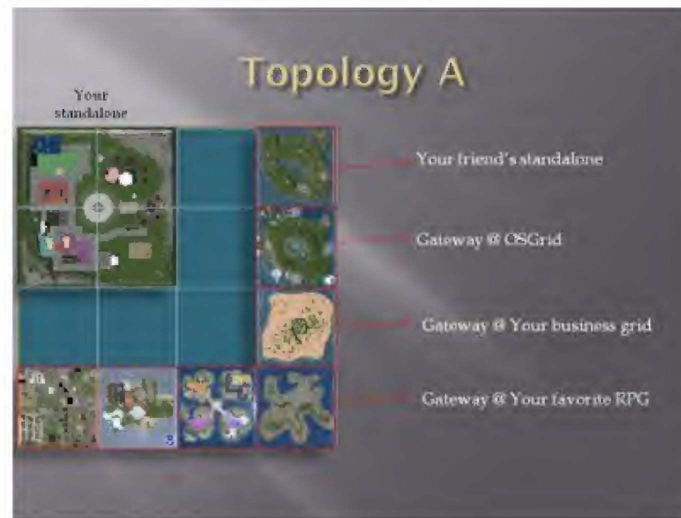
Los siguientes son algunos escenarios de uso. No hay una separación clara entre estos escenarios, hay un gran solapamiento entre ellas. Esto tampoco es una lista exhaustiva. El propósito de estas descripciones es darle algunas ideas de partida de cómo utilizar el hypergrid en la práctica.

Por favor, siéntase libre de añadir otros interesantes a esta lista.

Mundo personal

Esto se refiere al primer escenario opensims independiente (standalone). Normalmente, standalones están completamente desconectados de la Internet. Sin embargo, cuando se ejecuta en modo hypergrid, los standalones se convierten en el poder de la red. Como tal, usted puede ejecutar su propio mundo en su propio ordenador, y vincular su mundo a quien tú quieras. Por ejemplo, puede vincular a opensims de tus amigos y hacer hypergrid en grid abiertas como OSGrid.

Lo bueno de este escenario es que todos sus activos se almacenan en su ordenador, y no en otra persona servidor.



Comunidades.

Este segundo escenario es sobre las comunidades, en sentido amplio. La idea aquí es que un grupo de personas se reúnen para apoyar una red pequeña comunidad, es decir, un mundo común, donde las actividades tienen lugar para compartir. Pero al mismo tiempo, los miembros de la comunidad mantener sus propios mundos independientes. El standalones enlaza a la red de la comunidad y la grid de la comunidad puede tener un enlace de vuelta a los mundos individuales de los miembros y otros lugares de interés.

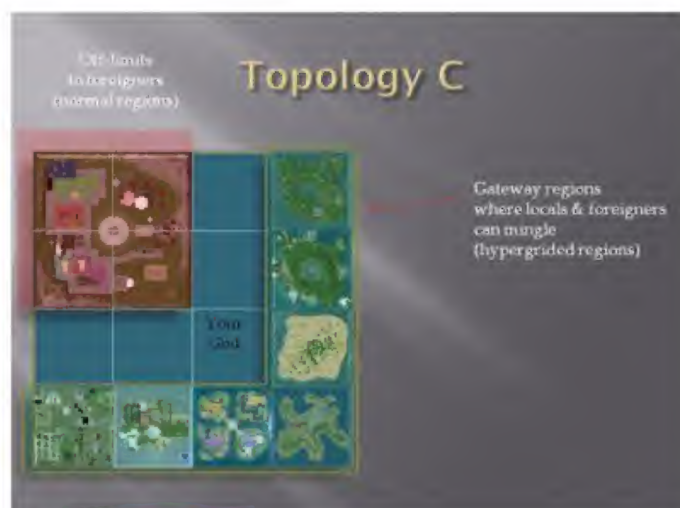
La identidad de los miembros son probablemente las identidades que tienen en sus standalone (en sus parcelas), y sus activos también son probablemente almacenados allí. Los activos presentes en las regiones de la comunidad, sin embargo, se almacenan en el servidor grid activo.



Red pública de Regiones.

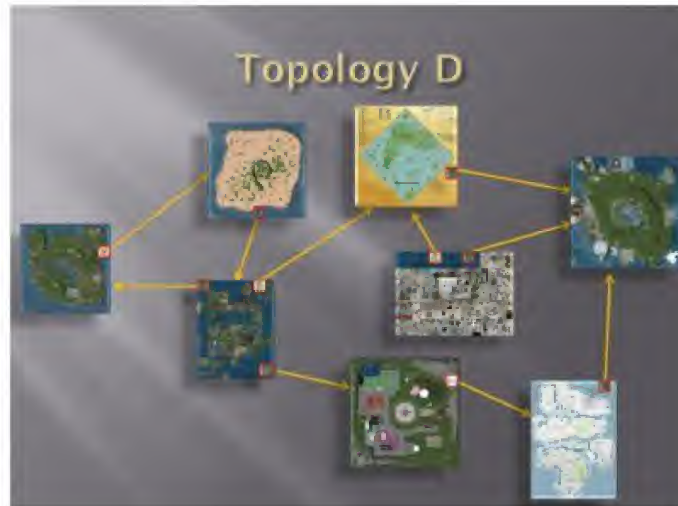
En este hibrido modo, hay una puerta de entrada para los usuarios locales y visitantes a la red para satisfacer arbitraria. Esta es también una buena estrategia para atraer nuevos usuarios a la red, ya que los usuarios al azar pueden visitar las regiones de puerta de enlace, sin tener que registrarse en una cuenta por adelantado.

Este modo híbrido es muy similar a lo que sucede en la web. Por ejemplo, cualquier persona puede visitar las páginas públicas de Facebook sin tener que registrarse para obtener una cuenta de Facebook. Sin embargo, sólo los usuarios de Facebook pueden ir más allá en su interior.



Nivel de juegos.

La versión normal de OpenSim cumplir un mapa común para todas las regiones en una rejilla. El hypergrid elimina esa restricción. Como tal, se convierte en fácil de diseñar juegos en el mundo se ve diferente dependiendo de donde está el jugador.



7.14.1. Instalar y correr un Hypergrid

Vamos a instalar y correr un Hypergrid. Vemos dos posibilidades:

1. Si estamos corriendo el OpenSim en el modo **GridMode** con servidores y otras máquinas, sólo necesitamos modificar el OpenSim.ini para habilitar el HyperGrid.

    - Abrimos el fichero OpenSim.ini:



    - Vemos que el gridmode esta a true que indica que conectamos nuestras regiones a una grid:

```
; Set this to true if you are connecting your regions to a grid
; Set this to false if you are running in standalone mode
gridmode = false
```

    - Fijamos la variable hypergrid a true porque queremos tener la funcionalidad de hypergrid:

```
; Set this to true if you want Hypergrid functionality
hypergrid = true
```

2. Si se está utilizando el opensim en independiente (standalone) y desea que sea útil desde la red, fije la variable hypergrid a true:

```
; Set this to true if you want Hypergrid functionality
hypergrid = true
```

Ejemplo de la Sección Network (red).

Vamos a mostrar ejemplos dependiendo de cómo este configurado nuestro OpenSim para configurarlo, todas estas configuraciones se hacen en nuestro fichero OpenSim.ini.

- Ejemplo por defecto, configuración de Network Settings para individual (Stand Alone):

```
[Network]
http_listener_port = 9000

grid_server_url = "http://example.com:9000"

grid_send_key = null
grid_recv_key = null

user_server_url = "http://example.com:9000"

user_send_key = null
user_recv_key = null

asset_server_url = "http://example.com:9000"

inventory_server_url = "http://example.com:9000"
messaging_server_url = "http://example.com:9000"
```

- Ejemplo por defecto, configuración Network para un Grided Server:

```
[Network]
http_listener_port = 9000

grid_server_url = "http://example.com:8001"
grid_send_key = null
grid_recv_key = null

user_server_url = "http://example.com:8002"
user_send_key = null
user_recv_key = null

asset_server_url = "http://example.com:8003"
inventory_server_url = "http://example.com:8003"
messaging_server_url = "http://example.com:8006"
```

- Ejemplo por defecto, configuración Network para OSGrid Attched Servers:

```
[Network]
http_listener_port = 9000
grid_server_url = "http://osgrid.org:8001" ;default 8003
grid_send_key = "1234"
grid_recv_key = "1234"

user_server_url = "http://osgrid.org:8002"
user_send_key = "1234"
user_recv_key = "1234"

asset_server_url = "http://assets.osgrid.org:8003"
inventory_server_url = "http://osgrid.org:8004" ;default 8003
messaging_server_url = "http://osgrid.org:8006"
```

Modificaciones que necesitan ser aplicadas al OpenSim.ini para habilitar el Hypergrid.

Vamos a habilitar el hypergrid.

- Cambia la siguiente línea de código en elOpenSim.ini:

```
hypergrid = false to true
```

- Añadir o modificar la sección ### WORLD MAP como se muestra:

```
; ##
; ## WORLD MAP
; ##

WorldMapModule = "WorldMap"
MapImageModule = "MapImageModule"
WorldMapModule = "HGWorldMapModule" ;NOT INCLUDED IN DEFAULT OpenSim.ini.EXAMPLE
```

Información: HGWorldMapModulel, después de teletransportate a otra grid, limpia los azulejos en tu mapa que dejaste de la grid en la cual tu estabas previamente.

- Modifica la sección [Archictecture]:

```
[Architecture]
;Include-Standalone = "config-include/Standalone.ini" ; DEFAULT UNCOMMENTED
Include-HGStandalone = "config-include/StandaloneHypergrid.ini"
;Include-Grid = "config-include/Grid.ini"
;Include-HGGrid = "config-include/GridHypergrid.ini"
```

Tu instancia de OpenSimimulator debería ahora estar preparada para usar Hypergrid.

Nota importante: Estate seguro que tu tienes fijado un 'home'. Si tu región home no existe, el hyperlink no podrá trabajar. Para fijar tu home, ve una vez a tu región local y 'set Home' desde tu visor.

Enlazar regiones

Esta es la sentencia que debemos de poner en nuestra consola si queremos enlazar regiones.

link-region <Xloc> <Yloc> nombre de dominio / direccion ip y el http_listener_port

- **Xloc y Yloc**.

  Utilice Xloc y Yloc que tienen sentido a tu mundo, es decir, cerca a sus regiones, pero no adyacentes.

- **http_listener_port**

  Del simulador donde la región está corriendo y tu quieres hacerle un enlace.

Tú puedes hacer un enlace a una específica región usando el nombre de la región al final.

Por ejemplo:

```
link-region 997 997 osl2.nac.uci.edu:9006:UCI Welcome
```

Desenlazar Regiones.

En la consola escribe por ejemplo:

```
unlink-region osl2.nac.uci.edu:9006

or

unlink-region <local region name>
```

7.14.2. Prueba de enlazar regiones externas a nuestro servidor

Para poder tener la capacidad de saltar entre regiones como si estuvieran en nuestro servidor.

Vamos a enlazar a un grid externo unas coordenadas locales. Pero tenemos que saber cuáles son estas coordenadas locales de nuestra máquina, para poder enlazar grid externos.

- Hacemos clic en el fichero opensim.ini:



- Vemos que la funcionalidad hypergrid esta apagada:

```
; Set this to true if you want Hypergrid functionality
hypergrid = false
```

- Activamos la funcionalidad hypergrid:

```
; Set this to true if you want Hypergrid functionality
hypergrid = true
```

- Vemos la sección Network:

```
grid_server_url = "http://127.0.0.1:8003"
grid_send_key = "null"
grid_recv_key = "null"

user_server_url = "http://127.0.0.1:8002"
user_send_key = "null"
user_recv_key = "null"

asset_server_url = "http://127.0.0.1:8003"

inventory_server_url = "http://127.0.0.1:8003"
```

- Configuramos la sección Network de la siguiente manera (tenemos encuenta que esta configuración es cuando estamos corriendo el OpenSim en modo StandAlone -independiente-):

```
grid_server_url = "http://192.168.0.16:9000"
grid_send_key = "null"
grid_recv_key = "null"

user_server_url = "http://192.168.0.16:9000"
user_send_key = "null"
user_recv_key = "null"

asset_server_url = "http://192.168.0.16:9000"

inventory_server_url = "http://192.168.0.16:9000"
```

- Habilitamos el WORLD MAP, para así poder ver en el mapa al otro metaverso:

```
; ##
; ## WORLD MAP
; ##

;WorldMapModule = "WorldMap"
;MapImageModule = "MapImageModule"
```

- Configuramos el WORLD MAP de la siguiente manera:

```
; ##
; ## WORLD MAP
; ##

WorldMapModule = "WorldMap"
MapImageModule = "MapImageModule"
WorldMapModule = "HGWorldMapModule"
```

- Vemos la arquitectura que tenemos definida es la Standalone, es decir, de OpenSim de forma individual:

```
[Architecture]
    ; Choose exactly one and only one of the architectures below.

    Include-Standalone    = "config-include/Standalone.ini"
    ;Include-HGStandalone = "config-include/StandaloneHypergrid.ini"
    ;Include-Grid         = "config-include/Grid.ini"
    ;Include-HGGrid       = "config-include/GridHypergrid.ini"
```

- Configuramos la arquitectura para que se ejecuta de manera Standalone con Hypergrid:

```
[Architecture]
    ; Choose exactly one and only one of the architectures below.

    ;Include-Standalone    = "config-include/Standalone.ini"
    Include-HGStandalone = "config-include/StandaloneHypergrid.ini"
    ;Include-Grid          = "config-include/Grid.ini"
    ;Include-HGGrid        = "config-include/GridHypergrid.ini"
```

- Escribimos en la consola del servidor show región:

```
Region (UC3M) # show region
```

- Nos muestra el siguiente resultado:

```
15:00:18 - Region Name: UC3M , Region XLoc: 9927 , Region YLoc: 4308 , Region Po
rt: 9000
```

- Una vez conocidas las coordenadas locales XLoc, YLoc de nuestro Sim, tenemos que enlazarlo al exterior.

- Hay que seleccionar un grid del listado público de grid, para eso vamos a la siguiente dirección web:

  http://opensimulator.org/wiki/Public_Hypergrid_Nodes

- Vemos los Sim que están activos para hacerles un hypergrid:

Hypergrid-enabled sims

link-region server and port	Organization	Description	Grid Location
osl3.nac.uci.edu:9000	University of California, Irvine	The "UC Irvine" region connected to OSGrid, neighboring Wright Plaza. You can link to it as a way to link to OSGrid.	OSGrid is centered at 10,000, 10,000
osl2.nac.uci.edu:9006	University of California, Irvine	The "UCI Welcome" region connected to OSGrid. It is run by Diva (Crista Lopes) on a machine owned by the University of California, Irvine. You can link to it as a way to link to OSGrid.	OSGrid is centered at 10,000, 10,000
osl3.nac.uci.edu:9000	University of California, Irvine	The "UC Irvine" region connected to OSGrid, neighboring Wright Plaza. You can link to it as a way to link to OSGrid.	OSGrid is centered at
condensationland.com:9000	Zonja Capalini	Condensation Land Grid, 5 regions and links to several other HG-enabled sims.	Centered at 7789, 7789

- Hay que tratar de enlazar grids que estén cerca de nosotros (en coordenadas x, y) con el fin de evitar errores por estar distantes.

- Vamos a seleccionar el grid osl3.nac.uci.edu:9000 y tenemos que enlazarlo utilizando coordenadas relativas a nuestro emplazamiento local, no tenemos que poner unas coordenadas que estén pegadas a nuestra:

```
Region (UC3M) # link-region 9928 4307 osl3.nac.uci.edu:9000
```

- Nos da la siguiente información indicándonos que estamos demasiado lejos:

```
15:48:01 - >> MapName: osl3.nac.uci.edu:9000

15:48:01 - [HGrid]: Link to osl3.nac.uci.edu:9000, in 2541568-1102592
15:48:01 - [HGrid]: Linking remote region osl3.nac.uci.edu:9000
15:48:01 - [HGrid]: Linking to http://128.200.80.232:9000/
15:48:02 - >> HERE, uuid: 726ca3e0-a51d-11dd-ad8b-0800200c9a66

15:48:02 - >> HERE, realHandle: 10996215791947776

15:48:02 - [HGrid]: Successfully linked to region_uuid 726ca3e0-a51d-11dd-ad8b-0
800200c9a66
15:48:02 - [HGrid]: Failed getting/storing map image, because it is probably alr
eady in the cache
15:48:02 - [HGrid]: Unable to link, region is too far (10001, 10000)
```

- Cambiamos las coordenadas de nuestra región:

```
Location = 10001,10001
```

- Ahora que estamos más cerca de osl3.nac.uci.edu:9000 le hacemos un link a ese metaverso, pero teniendo cuidado de no poner las coordenadas donde queremos verle en el mapa que estén pegadas a nuestra isla, es decir, nosotros estamos en las coordenadas 10001 y 10001, y decimos que queremos ver a osl3 en nuestro mapa en las coordenadas 1001, 10003:

```
Region (UC3M) # link-region 10001 10003 osl3.nac.uci.edu:9000
```

- Vemos la configuración de nuestro navegador, y vemos que apunta a nuestra dirección IP interna 192.168.0.16, y utiliza el puerto 9000:

Destino: anguage es -loginuri http://192.168.0.16:9000

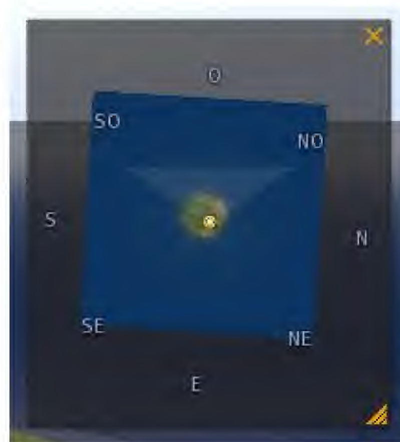
- Nos conectamos al metaverso UC3M:



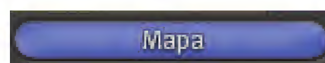
- Hacemos clic en el botón minimapa:



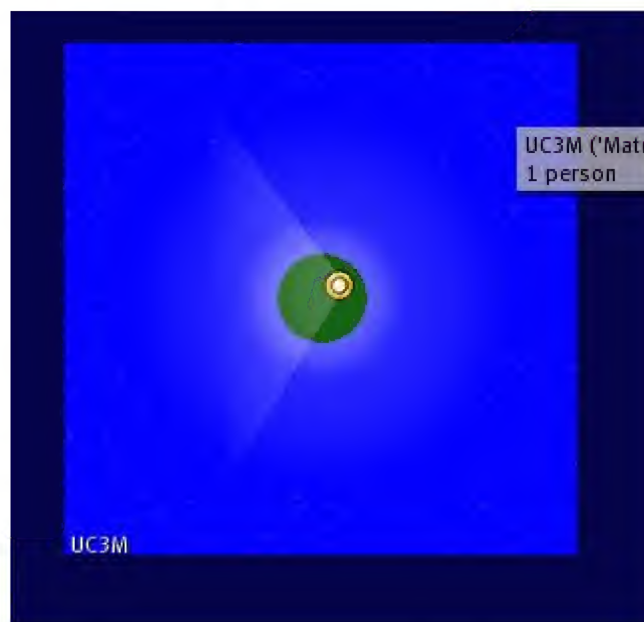
- Vemos el minimapa:



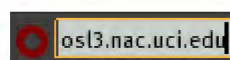
- Hacemos clic en el botón Mapa:



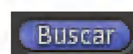
- Vemos el mapa:



- Introducimos nombre del metaverso que acabamos de buscar, que es osl3.nac.uci.edu:



- Hacemos clic en el botón **Buscar**:



- Vemos en el mapa el metaverso osl3.nac.uci.edu:



Figura 7.16. Metaverso osl3.nac.uci.edu.

Dependiendo de las coordenadas locales donde enlazamos, veremos la región condensationland.com en el mapa, y podemos teletransportarnos haciendo doble clic sobre el lugar elegido en el mapa.

- Hacemos clic en una posición del mapa:



- Vemos que nos sale una flecha indicándonos la posición:



- Hacemos clic en el botón **Teleport**:



- Vemos que no nos podemos teletransportarnos a ese metaverso porque no tenemos creado un usuario registrado en ese metaverso:



- Vemos que la flecha nos indica el lugar donde se encuentra el metaverso osl3:



- Miramos el minimapa y vemos que nos apunta la dirección donde esta el metaverso osl3:



- Ahora que nos desconectamos del metaverso UC3M, deshacemos el link:

```
15:58:03 - [HGrid]: Successfully unlinked osl3.nac.uci.edu:9000
```

7.15. Conectarse a OsGrid

7.15.1. Crear una cuenta en OsGrid

OsGrid permite enlazar varios ordenadores que tienen instalado OpenSim e interactuar ente ellos e interactuar también con otros mundos virtuales.

Para entrar en OsGrid, hay que crear una cuenta y tener instalado un explorador de mundos virtuales.

- Vamos a la siguiente dirección web:

  http://www.osgrid.org/elgg/

- Hacemos clic en el siguiente botón:

Register

- Nos aparece la siguiente panel para logearse:

Log in

Username

Password

Log in

Remember me

Register | Lost password

- Pulsamos sobre **Register** para crear una cuenta:

Register

- Nos sale la siguiente ventana:

Register

Account Name

giaa

Account Surname

Quixote

Email address

e100048464@gmail.com

Password

****

Password (again for verification)

****

Register

Figura 7.17. Crear cuenta OsGrid

Introducimos los datos que queremos que tenga nuestro avatar en OsGrid, es recomendable dar los mismos datos que tengamos en nuestra cuenta de Second Life, para posible unión de mundos virtuales en el futuro.

7.15.2. Elegir visor para OsGrid

Vamos a elegir un visor para entrar en OsGrid.

- Entramos en nuestro correo electrónico, y OsGrid nos manda a la siguiente página Web:

  http://osgrid.org/docs/instructions.htm

- La página web nos muestra la siguiente información:

Hippo OpenSim Viewer (*recommended)

osgrid.org does not provide technical support for any viewers

Step 1 :: Create a grid account on the menu.

Step 2 :: Download the Hippo Viewer.

(Defaults to OSGrid supports systems listed below)

Windows Viewer : download (osgrid.org) 8.13mb (exe)

OSX Viewer : download (osgrid.org) 63.5mb (dmg)

Linux Viewer : download (osgrid.org) 12.97mb (zip)

Step 3 :: Start the Hippo viewer from your newly created icon and enter your Firstname, Lastname and Password that you chose and connect to the grid.

Figura 7.18. Recomendación de visor para OsGrid.

Nos recomiendan que utilicemos el cliente (visor) “Hipo OpenSim”.

7.15.2.1. Utilizar el visor de Linden Lab

Linden Labs Second Life Viewer

osgrid.org does not provide technical support for any viewers

Step 1 :: Create a grid account on the menu.
Download the Client from Linden Labs and install it. (continue to step

Step 2 :: Configure your launcher or shortcut according to your Operating system of choice.

Microsoft Windows

A. Make a copy of the Second Life icon on your desktop

B. Right click it copy and then right click your desktop and paste it and rename the icon to OSGRID

C. Then right click it and edit the Shortcut.

Look for the Path and enter after SecondLife.exe the following code:

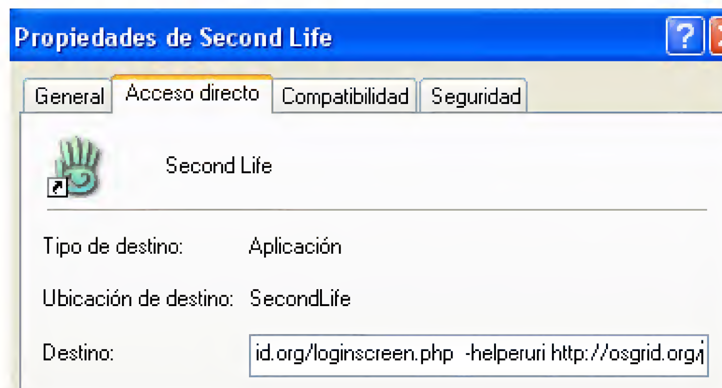
```
-loginuri http://osgrid.org:8002 -loginpage http://osgrid.org/loginscreen.php  -helperuri http://osgrid.org/
```

example:

```
"C:\Program Files\SecondLife\SecondLife.exe"  -multiple -loginuri http://osgrid.org:8002  -loginpage  http://osgrid.org/loginscreen.php -helperuri http://osgrid.org/
```

Si utilizamos el cliente (visor) de Linden Lab, tenemos que seguir los siguientes pasos:

- En el visor de Lindel Lab, tenemos que dar al botón derecho, y pulsamos sobre propiedades:



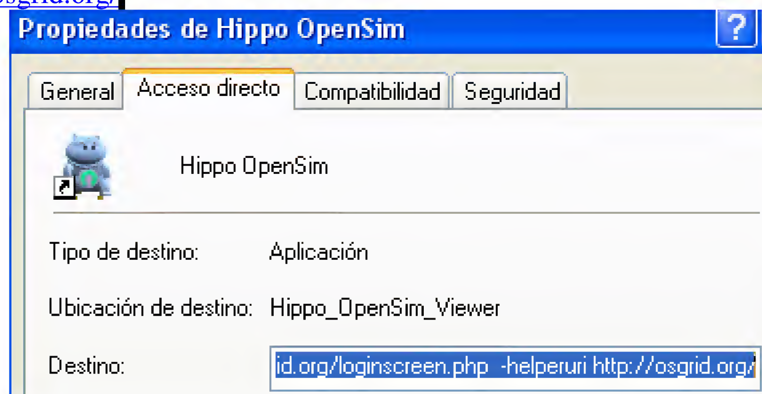
En la ficha "Acceso directo", en el apartado Destino, introduciremos a continuación **-loginuri** http://osgrid.org:8002 **-loginpage** http://osgrid.org/loginscreen.php **-helperuri** http://osgrid.org/

7.15.2.2. Utilizar el visor de Hippo.

En el visor de Lindenl Lab, tenemos que dar al botón derecho, y pulsamos sobre propiedades.

- Nos colocamos en la pestaña de "Acceso directo", y en la región de destino introducimos la siguiente información:

loginuri http://osgrid.org:8002 -loginpage http://osgrid.org/loginscreen.php -helperuri http://osgrid.org/



7.15.3. Entrar en OsGrid

Vamos a entrar en OsGrid.

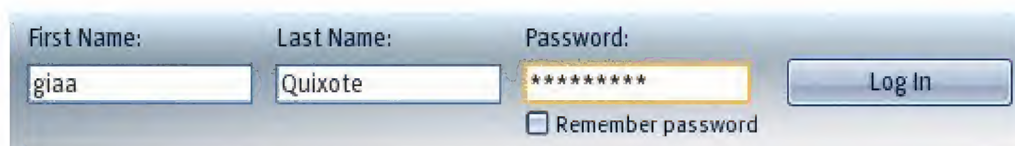
- Abrimos el explorador de Hippo:



- Seleccionar un sitio donde ir en OsGrid:



- Introducimos los datos en el explorador de Hippo:



- Ya estamos dentro de OsGrid:



Figura 7.19. Avatar como nube

Nos damos cuenta que nuestro avatar es una nube, pues todavía no le hemos dado forma humana.

- Exploramos OsGrid:



Figura 194. El metaverso OsGrid

Como podemos apreciar OsGrid no tiene nada que envidiar a Second Life, y además es un mundo virtual completamente gratuito.

7.15.4. Dar forma humana al avatar

Vamos a dar forma humana al avatar.

- Para conseguir tener forma humana, tenemos hacer clic en objeto siguiente:



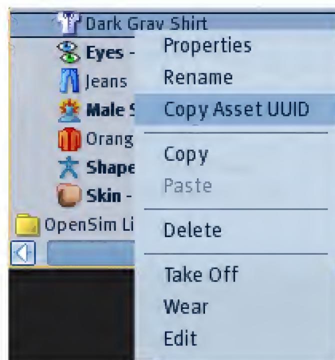
Figura 7.21. Cuerpo del avatar en OsGrid

Este objeto nos ofrece un cuerpo y ropa gratuitamente, y nosotros la aceptamos.

- <u>Ahora miramos nuestro inventario</u>:



- <u>Seleccionamos cada uno de los objetos del inventario y damos a **wear**</u>:



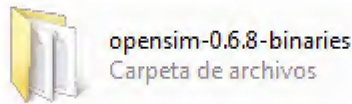
- <u>Ya hemos dado forma humana al avatar</u>:



7.16. Enlazar Sim de OpenSim a OsGrid

Vamos a enlazar nuestra Sim de OpenSim a OsGrid.

- Accedemos a nuestra carpeta "opensim-0.6.8:



- Abrimos el archivo OpenSim.ini que está en la carpeta bin:



- Vemos que el modo grid esta desconectado:

```
; Set this to true if you are connecting your regions to a grid
; Set this to false if you are running in standalone mode
gridmode = false
```

- Conectamos el modo grid:

```
gridmode = true
```

- Vamos a la sección Network:

```
[Network]

grid_server_url = "http://127.0.0.1:8003"
grid_send_key = "null"
grid_recv_key = "null"

user_server_url = "http://127.0.0.1:8002"
user_send_key = "null"
user_recv_key = "null"

asset_server_url = "http://127.0.0.1:8003"

inventory_server_url = "http://127.0.0.1:8003"

; The MessagingServer is a companion of the UserServer. It uses
; user_send_key and user_recv_key, too
messaging_server_url = "http://127.0.0.1:8006"
```

- Introducimos la IP de OsGrid http://osgrid.org y las claves de osgrid 1234:

```
grid_server_url = "http://osgrid.org:8003"
grid_send_key = "1234"
grid_recv_key = "1234"

user_server_url = "http://osgrid.org:8002"
user_send_key = "1234"
user_recv_key = "1234"

asset_server_url = http://osgrid.org:8003"

inventory_server_url = "http://osgrid.org:8003"
```

- Accedemos a la carpeta de la región:



- Abrimos el archivo Regions.ini:



- El fichero regions.ini contiene la siguiente información:

```
[tesis]
RegionUUID = 918ecb15-03e8-478d-bcb3-e3370e0cbfcd
Location = 1000,1000
InternalAddress = 0.0.0.0
InternalPort = 9000
AllowAlternatePorts = False
ExternalHostName = 127.0.0.1
MasterAvatarFirstName = giaa
MasterAvatarLastName = Quixote
MasterAvatarSandboxPassword =
```

7.16.1. Cambiar la localización del fichero región.ini

Vamos a buscar una localización que este libre y que este cerca de otras regiones:

- Vamos a la página web de OsGrid:

http://www.osgrid.org/elgg/pg/utilities/connect

- Vemos la **lista de regiones**:

http://www.osgrid.org/elgg/pg/utilities/regions

- Vamos al **mapa**:

http://www.osgrid.org/elgg/pg/utilities/opencoordinates

- Vemos que las coordenadas del mapa:

X	Y	Distance	Neighbour Count	Neighbours
9987	10004	362 m	7	Bade Plaza, Blue Ridge Estates, Open Market, Abyss, Port Bade, RezzMe, Kartengeheimnis
9988	10001	512 m	7	Space Place, Auspicious, Burroughs, Moebius, Lippo, Astonia, Virtual Wallet
9987	10001	572 m	6	Burroughs, Sandbox Island, Lippo, show info, Astonia, Virtual Wallet
9987	10005	572 m	7	Blue Ridge Estates, Open Market, Port Bade, Delirium City, Thor's Realm, Kartengeheimnis, Moonglow

- Vamos a elegir coordenadas que estén cerca de una región y las introducimos en default.xml:

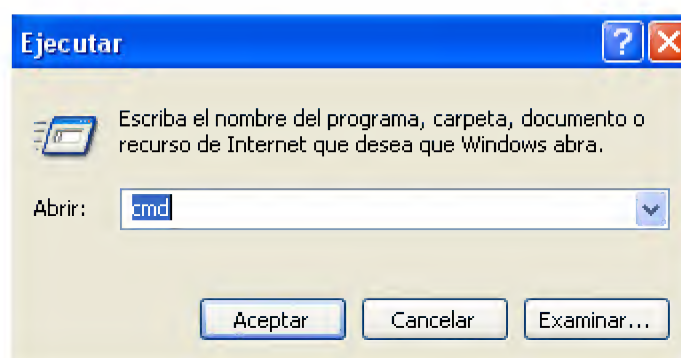
```
Location = 9987,10009
```

7.16.2. Cambiar las conexiones de red del fichero region.ini

Vamos a cambiar las conexiones de red del fichero de region.ini.

- Para saber los datos de red de nuestro ordenador hacemos clic en el botón inicio/ejecutar:

- Dentro de la ventana ejecutar escribimos el comando **cmd**:



- Ponemos en la consola el siguiente comando **ipconfig**:

```
C:\>ipconfig

Configuración IP de Windows


Adaptador Ethernet Conexión de área local 2          :

        Estado de los medios. . . .: medios desconectados

Adaptador Ethernet Conexiones de red inalámbricas 2          :

        Sufijo de conexión específica DNS : uc3m.es
        Dirección IP. . . . . . . . . . . : 163.117.85.36
        Máscara de subred . . . . . . . . : 255.255.248.0
        Puerta de enlace predeterminada   : 163.117.80.2
```

- Introducimos en default.xml:

  - **internal_ip_address**:

```
internal_ip_address="163.117.85.36"
```

  - **external_host_name**:

```
external_host_name="163.117.85.36"
```

Para saber la external_host_name vamos a la siguiente dirección:

http://www.osgrid.org/elgg/pg/utilities/connect

- Ya hemos conectado nuestra región Sim a OsGrid:

Region:	X	Y
TESIS	9987	10009
Vette5	9987	9998
Helene	9987	9988
Telesto	9987	9987
Starlight Spaceport	9987	9985
Vette6	9987	9999
Astonia	9987	10000
HCMServerLand	9987	10007

Figura 7.22. Tesis en OsGrid

Podemos apreciar que Tesis aparece en OsGrid.

- Recordar que la consola de Tesis tiene que estar encendida, si queremos que nuestro OpenSim este operativo:

```
Region (TESIS) #
```

7.16.3. Acceder a nuestra isla conectada a OsGrid

Vamos a acceder a nuestro Sim que está conectado a OsGrid.

- Accedemos a nuestra isla Tesis en OsGrid:



Figura 7.23. Acceso de Tesis en OsGrid

Comprobamos que nuestro avatar es una nube.

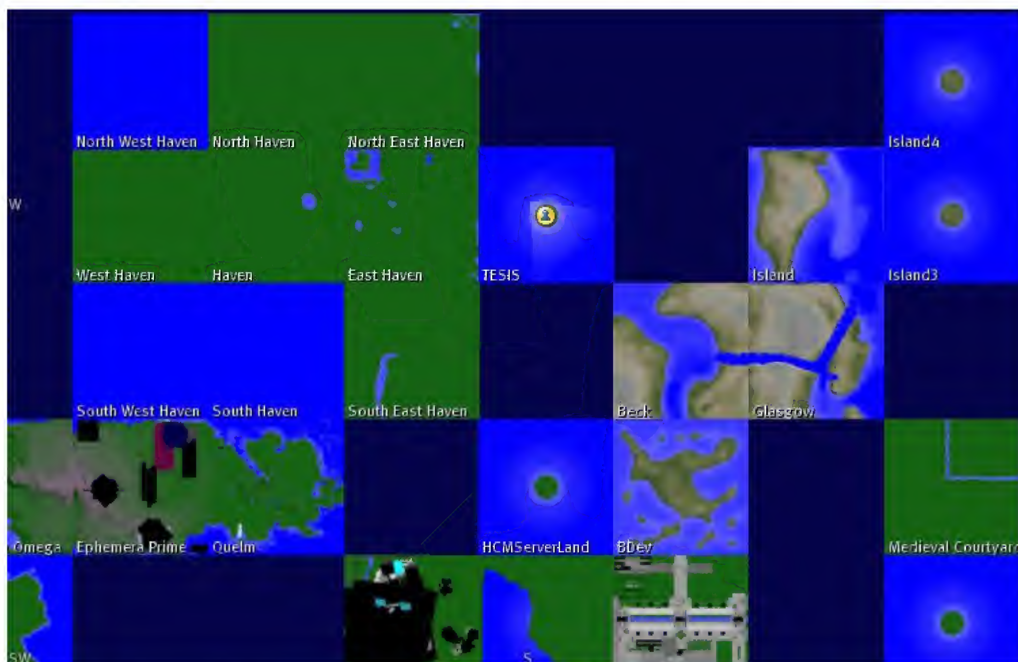
- Entramos en OsGrid y le ponemos un cuerpo y ropa al avatar, y después me he transportado a mi isla TESIS en OsGrid:



Figura 7.24. Avatar en Tesis conectada a OsGrid

Podemos comprobar que el avatar tiene un cuerpo y esta vestido en nuestra isla Tesis en OsGrid.

- Comprobamos el mapa de OsGrid y vemos que TESIS aparece:



- Comprobamos el teletransporte a otras islas.

En TESIS ahora no tenesmos conexión con otras islas, con lo cual no puedo teletransportarme a otras islas, esto es debido a que tenemos que tener nuestra isla TESIS

más tiempo conectada a OsGrid, para que las nuestra isla se comunique con las demás islas y pueda teletransportarme.

7.17. Realxtend

RealXtend es una tecnología basada en OpenSim, han integrado módulos dentro de opensim para mejor los gráficos y es una tecnología completamente gratuita.



Las mejoras que ofrece son:

- Trata al avatar como una entidad universal, de esta forma podemos navegar desde un metaverso a otro metaverso con el mismo avatar.

- Mejora de los gráficos con respecto a Second Life.

- Posibilita la utilización de herramientas de estudio en 3D para subir objetos diseñados con estas herramientas al metaverso creado con realxtend.

  Ejemplo.: Podemos modelar un sofá con una herramienta de estudio en 3D y subir el sofá al metaverso de realxtend.

7.17.1. Teletransportarse entre Metaverso UC3M y Second Life.

El visor RealXtend habilita el teletransporte entre OpenSim y Second Life, por lo tanto vamos a teletransportarnos desde el metaverso UC3M a Second Life.

El visor de realxtend vale tanto para second life, opensim y realxtend. Si queremos entrar en un metaverso creado en realxtend necesitamos el visor de realxtend.

- Nos tenemos que descargar el visor RealXtend, desde la siguiente dirección web:

  http://www.realxtend.org/

- Hacemos clic en el siguiente enlace:



- Nos aparece el siguiente enlace para bajarnos el RealXTend:

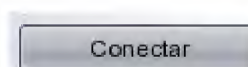
realXtend Viewer 0.42 (86 Mb)
Download realxtendviewer_0.42_setup.exe from realxtend.org

- Descargamos el programa, y lo instalamos, y nos crea el siguiente icono, en el cual hacemos clic:



- Nos aparece la siguiente información:

- Introducimos la dirección IP y el puerto de nuestra computadora en el siguiente cuadro de texto:

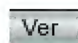


- Hacemos clic en el botón **conectar**:

- Aparece nuestro avatar giaa Quixote en el metaverso UC3M:



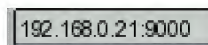
- Hacemos clic en la pestaña ver del navegador:



- En el submenú que nos sale elegimos la opción **Address Bar**:



- Vemos que aparece la barra de navegación:



- Introducimos el nombre de **Second Life**:



- Nos aparece le siguiente cuadro de texto para que nos teletransportemos a Second Life:

Teleport

Do you want to teleport to Second Life?

Input below the username & password you wish to use at the destination.

User Name: giaa Quixote

Password: ****************

Teleport   Cancel

- Y hacemos clic en el botón Teleport:



- Ya estamos en Second Life:



- Ahora introducimos la dirección del metaverso UC3M. Nos aparece el siguiente cuadro de texto:

192.168.0.21:9000

Teleport

Do you want to teleport to 192.168.0.21:9000/?

Input below the username & password you wish to use at the destination.

User Name: giaa Quixote

Password: ****************

Teleport Cancel

- Y hacemos clic en el botón **Teleport**:



- Y ya estamos en el metaverso UC3M:



7.17.2. Creación de una cuenta de mundos virtuales realxtend y acceder a un mundo

Vamos a crear una cuenta para entrar en un mundo virtual de realxtend.

- Apunta el navegador web a la siguiente dirección:

http://users.realxtend.net/Register.aspx

- Se nos abre la siguiente ventana:

Login Register

Register

Account:(*) @world.realxtend.org
First name:(*)
Surname:(*)
Avatar name:(*)
Email address:(*)
Confirm email address:(*)
Password:(*)
Confirm password:(*)
Avatar storage url: http://avatar.realxtend.org:10000
Skype ID:
Homeworld address: world.realxtend.org
REGISTER

- Rellene el siguiente formulario. Recuerde el nombre de cuenta y la contraseña que dar. También de su dirección de correo electrónico real.

Register

Account:(*) giaa @world.realxtend.org
First name:(*) eduardo
Surname:(*) rojo
Avatar name:(*) giaa
Email address:(*) e100048464@gmail.com
Confirm email address:(*) e100048464@gmail.com
Password:(*) •••••••••
Confirm password:(*) •••••••••
Avatar storage url: http://avatar.realxtend.org:10000
Skype ID: 1
Homeworld address: world.realxtend.org

- Cuando esté listo, pulse el botón Register. En caso de que algo va mal la forma volverá a aparecer con la información que acaba de escribir.

REGISTER

- Vemos que nos aparece la siguiente información después de registrarnos con éxito:

Thank you for registering!
You will shortly receive register confirmation email.
Please, click the link in the email to confirm your registration.

- Después de registrarse con éxito, recibirá un correo electrónico de confirmación. Haga clic en el enlace proporcionado en el correo electrónico.



» Register confirmation

- Recibimos un e-mail de confirmación, y hacemos clic en el enlace para confirmar que es nuestra cuenta de correo electrónico:

Register confirmation Recibidos | X

realXtend para usuario mostrar detalles 21:26 (4 minutos antes) Responder

Thank you for registering. To confirm Your registration click the link below: http://avatar.realxtend.org/ConfirmRegistration.aspx?account=giaa&regKey=facc1d35-1f2a-0468-c940-ca2ff253cf26 . realXtend

- Al hace clic en el enlace nos sale el siguiente mensaje:

Login Register

Thank you for registering! You can now login to realXtend or login to authentication web service to modify your profile.

- Felicitaciones, ahora tiene una cuenta en nuestro servidor! Usted no necesita repita estos primeros cuatro pasos cuando se visita una vez más los mundos virtuales.

- Ahora puede iniciar sesión. Inicie el visor de realXtend.



- Para escribir en el campo user name: youraccount@avatar.realxtend.org

User Name: giaa@avatar.realxtend.org

Figura 195. Tu cuenta en realxtend

Donde youraccount es el nombre de la cuenta que proporcionaste cuando te has registrado.

- Escribe en el campo Password tu contraseña de cuenta:

Contraseña: *********

- Escribe en el campo Connect to: world.realxtend.org

Connect To: world.realxtend.org

- Presiona el botón conectar:



NOTA: Si no se tiene Flash instalado, usted no puede ver todos los elementos. Usted todavía puede explorar los mundos normalmente. Usted puede obtener Flash desde el sitio web de Adobe.

7.17.3. Realxtend StandAlone

Vamos a crear un metaverso StandAlone de Realxtend.

- Vamos a la siguiente dirección web:



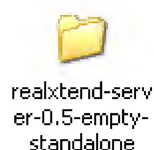
- Vemos el siguiente de descarga de realxtend en modo standalone, y hacemos clic en el enlace para descargarnos la aplicación:

realXtend Server 0.5 (80 Mb)
Download realXtend-server-0.5-empty-standalone.zip from realXtend.org

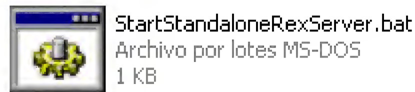
- Ya hemos descargado la aplicación para hacer un metaverso en realxtend en modo standalone:



- Descomprimimos la aplicación, y entramos en la carpeta:



- Hacemos clic en el programa StartStandaloneRExServer.bat:



- Hacemos clic en el acceso directo de realXtend para abrir el navegador:



- Introducimos el "**User Name**": testuser@127.0.0.1

User Name: testuser@127.0.0.1

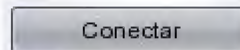
- Introducimos la contraseña:

Contraseña: ****************

- Seleccionamos en "**Connect To**" a "Authentication Server Home Grid":

Connect To: Authentication Server Home Grid

- Hacemos clic en el botón Conectar:



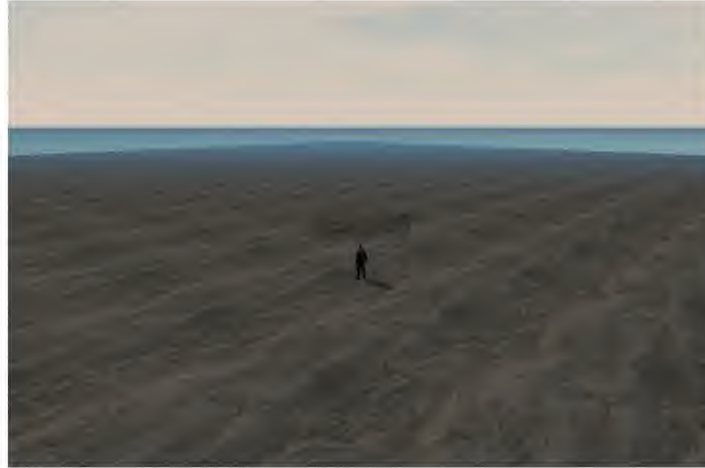
- Entramos en el metaverso de realxtend en modo standalone:



- Introducimos en la consola de región el comando **terrain fill 25**:

```
Region (ModreX) # terrain fill 25
```

- Vemos como se ha alisado el terreno:



7.17.4. Realxtend standAlone BeneathTheWaves

Vamos a ver como configurar la tecnología realxtend en modo standalone.

- Vemos los distintos tipos de servidores de realxtend:

realXtend Server 0.5 (80 Mb)
Download realXtend-server-0.5-empty-standalone.zip from realXtend.org
Download realXtend-server-0.5-BeneathTheWaves.zip from realXtend.org
Download realxtendserver_0.5.zip from SourceForge

- Hacemos clic en el siguiente enlace:

Download realXtend-server-0.5-BeneathTheWaves.zip

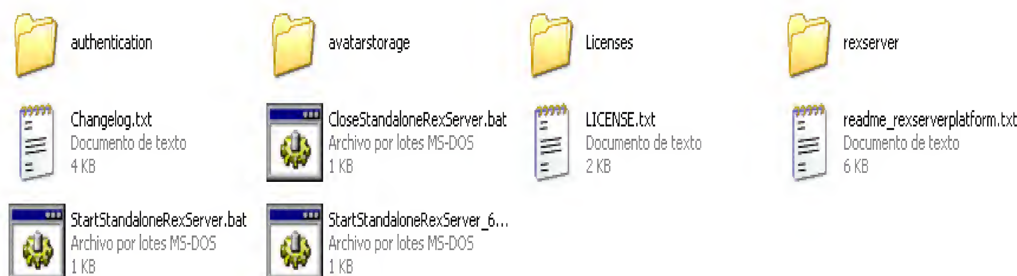
- Nos crea el siguiente fichero comprimido de realxtend-server-0.5:


realXtend-server-0.5-Beneath...
78.682 KB

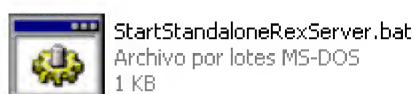
- Descomprimimos el fichero comprimido de realxtend-server-0.5:


realXtend-server-0.5-Beneath...

- Entramos dentro de la carpeta de realxtend-server-0.5:



- Hacemos clic en el siguiente fichero StartStandaloneRexServer.bat:



- Nos abre tres ventanas:

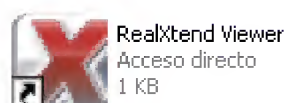
```
C:\Documents and Settings\Administrador\Escritorio\Nueva carpeta\realXtend-server-0.5-B...
dor\Escritorio\Nueva carpeta\realXtend-server-0.5-BeneathTheWaves\authentication
\Authentication.Module.AuthenticationManager.dll]
17:02:40 - [MODULES]:    [AuthenticationManager module]: Initializing.
17:02:40 - [MODULES]: Found Module Library [C:\Documents and Settings\Administra
dor\Escritorio\Nueva carpeta\realXtend-server-0.5-BeneathTheWaves\authentication
\Authentication.Module.FriendManager.dll]
17:02:40 - [MODULES]:    [FriendModule]: Initializing.
17:02:40 - [MODULES]: Found Module Library [C:\Documents and Settings\Administra
dor\Escritorio\Nueva carpeta\realXtend-server-0.5-BeneathTheWaves\authentication
\Authentication.Module.InstantMessaging.dll]
17:02:40 - [MODULES]:    [IM module]: Initializing.
17:02:40 - [MODULES]: Found Module Library [C:\Documents and Settings\Administra
dor\Escritorio\Nueva carpeta\realXtend-server-0.5-BeneathTheWaves\authentication
\Authentication.Module.UserManager.dll]
17:02:40 - [MODULES]:    [User module]: Initializing.
17:02:40 - [MODULES]: Could not load types for [AuthenticationEngine, Version=1.
0.0.0, Culture=neutral, PublicKeyToken=null].
17:02:41 - [MODULES]: Could not load types for [NHibernate, Version=2.0.1.4000,
Culture=neutral, PublicKeyToken=aa95f207798dfdb4].
17:02:41 - [MODULES]: The file [file:///C:\Documents and Settings\Administrador\
Escritorio\Nueva carpeta\realXtend-server-0.5-BeneathTheWaves\authentication\sql
ite3.dll] is not a module assembly.
17:02:41 - Enter help for a list of commands
17:02:41 - Authentication# :
```

```
C:\Documents and Settings\Administrador\Escritorio\Nueva carpeta\realXtend-server-0.5-B...
17:03:17 - [PRESENCE]: Failed to notify message server of agent location for 965
8af8d-5e78-464f-a216-88a30fda83e5
17:03:17 - [SCENE]: Could not find user info for 9658af8d-5e78-464f-a216-88a30fd
a83e5 when making it a root agent
17:03:17 - [REXAPR]: Sending user 9658af8d-5e78-464f-a216-88a30fda83e5 appearanc
e to all users. []
17:03:17 - [REXAPR]: Skipping blank URL on user...
17:03:17 - [REXAPR]: Sending user 9658af8d-5e78-464f-a216-88a30fda83e5 appearanc
e to all users. [http://127.0.0.1:10000/avatar/499a33362ab344358775e687bde2b648]

17:03:17 - [RexBotManager]: Added bot Istiophorus Platyperus to scene.
17:03:17 - [SUN]: Sun Settings Update: Fixed Sun? : False
17:03:17 - [SUN]: Sun Settings Update: Sun Hour   : 18.32833
17:03:17 - [SUN]: PosTime : 1271790197
17:03:17 - [RegionReady]: Disabled
17:03:17 - [RADMIN]: Creating default avatar entries
17:03:17 - [RADMIN]: Exception whilst loading default avatars ; Referencia a obj
eto no establecida como instancia de un objeto.
17:03:17 - [RADMIN]: Default avatars not loaded
17:03:17 - [!]:STARTUP COMPLETE
Currently selected region is Beneath the Waves
17:03:17 - [STARTUP]: Startup took 0m 36s
17:03:30 - [NHIBERNATE]: updating RexObjectProperties 0e0be1b6-731e-4bd2-90c4-e2
b77f9fac92
Region (Beneath the Waves) #
```

```
C:\Documents and Settings\Administrador\Escritorio\Nueva carpeta\realXtend-server-0.5-B...
17:02:41 - [REQUEST HANDLER]: Loaded database dll IDataBaseHandler Interface
17:02:41 - [REQUEST HANDLER]: Loaded authentication dll IAuthentication Interfac
e
17:02:41 - [HTTPD]: AVATARSTORAGE: Attempting to load PersonalInventoryService.d
ll
17:02:41 - [AGENTINVENTORY]: Inventorystorage: Attempting to load PersonalInvent
ory.SQLITE.dll
17:02:41 - [Inventory]: Sqlite - connecting: inventoryStore.db
17:02:41 - [DATASTORE]: Populated Inventory Folders Definitions
17:02:41 - [DATASTORE]: Populated Inventory Items Definitions
17:02:41 - [AGENTINVENTORY]: Added IInventoryData Interface
17:02:41 - [ASSET SERVER]: LoadDatabasePlugin: Attempting to load PersonalInvent
ory.SQLITE.dll
17:02:41 - [ASSET SERVER]: Added SQLite Asset storage engine 1.0.0.0
17:02:41 - [REST]: In Get Request
17:02:41 - [HTTPD]: AVATARSTORAGE: Loaded personal inventory plugin
17:02:41 - [HTTPD]: Starting up HTTP Server
17:02:41 - [HTTPD]: Listening port: 10000
17:02:41 - [HTTPD]: Server address: 127.0.0.1
17:02:41 - [HTTPD]: Starting http listener
17:02:41 - [AVATARSTORAGE]: Type "shutdown" + Enter to quit
```

- Hacemos clic en el visor de realxtend:



- Introducimos el nombre de usuario **testuser@127.0.0.1**:

User Name: testuser@127.0.0.1

Figura 7.26. Nombre de usuario de realxtend

Estamos corriendo el metaverso realxtend en modo standalone, por eso introducimos este nombre de usuario.

- <u>Introducimos la clave del usuario que es **test**</u>:

Contraseña: ****************

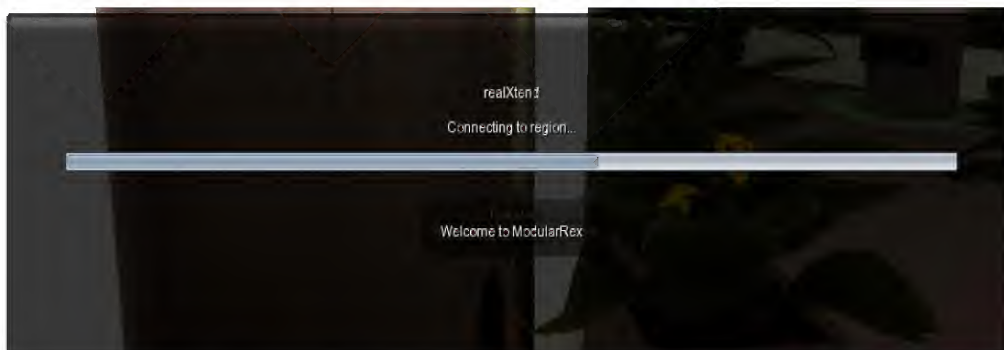
- <u>Y dejamos la conexión a **Authentication Server Home Grid**</u>:

Connect To: Authentication Server Home Grid

- <u>Hacemos clic en el botón **Conectar**</u>:

Conectar

- <u>Se carga nuestro metaverso</u>:



- <u>Entramos nuestro metaverso en modo standalone</u>:



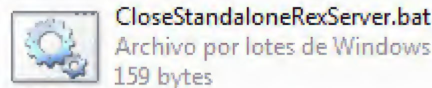
Figura 7.27. Metaverso realxtend en modo standalone.

Podemos comprobar que los gráficos son muy buenos, y que el avatar esta mejor diseñado.

- En esta imagen podemos apreciar la calidad de los gráficos de realxtend:



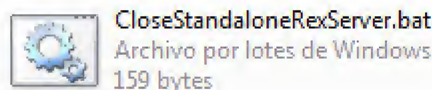
- Para cerrar el metaverso standalone de realxtend hacemos clic en el siguiente fichero:



7.17.5. Como configurar la plataforma del servidor para que otras personas puedan conectarse a él

Para habilitar la plataforma de servidor de LAN o el uso de Internet para que otras personas también pueden acceder a la Sim que hemos creado en modo standalone.

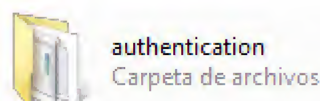
- Asegúrese de que todos los servidores están apagados.
- Si están encendidos hacemos clic en el fichero CloseStandaloneRexServer.bat:



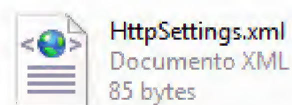
7.17.5.1. Configurando la autentificación

Vamos a configurar la autentificación.

- Abre la carpeta "**authentication**":



- Hacemos clic en el fichero HttpSettings.xml:

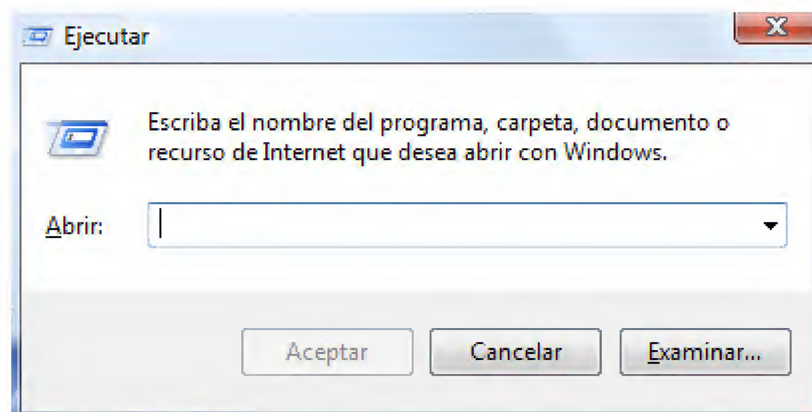


- Vemos que tiene el siguiente contenido:

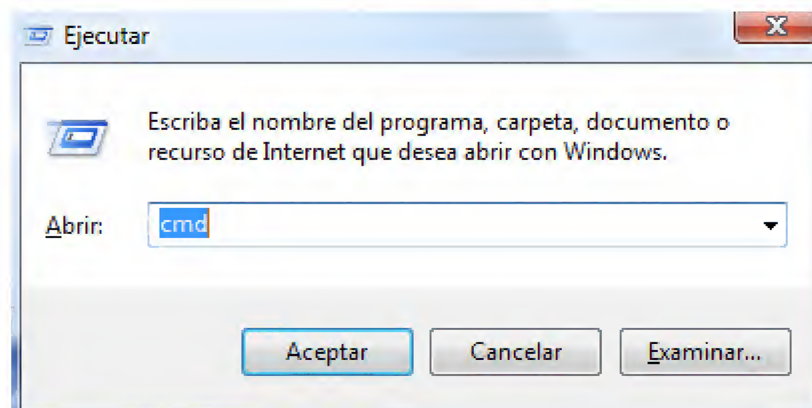
```
<HttpSettings>
  <address>127.0.0.1</address>
  <port>10001</port>
</HttpSettings>
```

- Fija la dirección de la externa IP de tu maquina.

- Si tú no sabes la dirección IP, haz esto: da simultáneamente la tecla de Windows + el botón R:



- En el cuadro de dialogo que se ha abierto escribe cmd y presiona enter:



- En la ventana que se abre, escribe el comando ipconfig y presina enter:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  Reservados todos los derechos.

C:\Users\100048464>ipconfig
```

- Vemos la siguiente información:

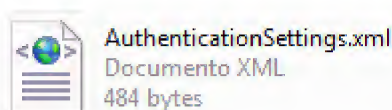
```
Adaptador LAN inalámbrico Conexión de red inalámbrica:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::e43e:6c4f:decc:b2bc%12
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.2.148
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.2.1
```

- Introducimos la dirección IP de nuestra máquina (estamos en una red LAN) y guarda el fichero:

```
<address>192.168.2.148</address>
```

- Abrimos authentication\AuthenticationSettings.xml:



- Vemos el contenido del fichero AuthenticationSettings.xml:

```
<AuthenticationSettings>
  <database_dll>SQLite_SqlHandler.dll</database_dll>
  <logging_enabled>False</logging_enabled>
  <default_Start_Region_X>1000</default_Start_Region_X>
  <default_Start_Region_Y>1000</default_Start_Region_Y>
  <default_avatar_storage_url>http://127.0.0.1:10000
</default_avatar_storage_url>
  <default_grid_url>http://127.0.0.1:9000</default_grid_url>
  <dns_server>
  </dns_server>
  <check_server_ip>True</check_server_ip>
</AuthenticationSettings>
```

- Fija el default_avatar_storage_url y default_grid_url a la IP externa de tu máquina. Deja los números de los puertos al final de las direcciones. Salva y cierra.

```
    <default_avatar_storage_url>http://192.168.2.148:10000
  </default_avatar_storage_url>
<default_grid_url>http://192.168.2.148:9000</default_grid_url>
```

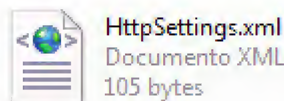
7.17.5.2. Configurar el almacén del avatar

Vamos a configurar la zona en la cual el avatar almacena su información.

- Abre la carpeta "**avatarstorage**":



- Abre el fichero HttpSettings.xml:



- Abrimos el fichero HttpSettings.xml:

```
<HttpSettings>
  <address>127.0.0.1</address>
  <port>10000</port>
  <ssl>false</ssl>
</HttpSettings>
```

- Fija la dirección IP externa de tu máquina. Salva y cierra:

```
<HttpSettings>
  <address>192.168.2.148</address>
  <port>10000</port>
  <ssl>false</ssl>
</HttpSettings>
```

7.17.5.3. Configurando el rexserver

- Abre la carpeta rexserver:



- Abrimos la carpeta regions:



- Abrimos el fichero default.xml:



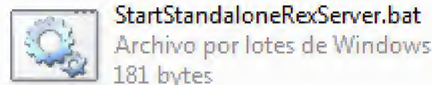
- Vemos el contenido de default.xml:

```
<Root>
  <Config sim_UUID="befb7ca2-beee-4469-9199-784ff918f3f9"
sim_name="Beneath the Waves" sim_location_x="1000"
sim_location_y="1000" internal_ip_address="0.0.0.0"
internal_ip_port="9000" allow_alternate_ports="false"
external_host_name="127.0.0.1" master_avatar_uuid="00000000-0000-0000-
0000-000000000000" master_avatar_first="Test"
master_avatar_last="User" master_avatar_pass="test"
lastmap_uuid="bc75b971-8ad3-4255-958f-0d9e117ec1c4"
lastmap_refresh="1271787981" nonphysical_prim_max="0"
physical_prim_max="0" clamp_prim_size="false" object_capacity="0" />
</Root>
```

- Cambia el parámetro external_host_name por la IP externa de tu máquina. Salva y cierra:

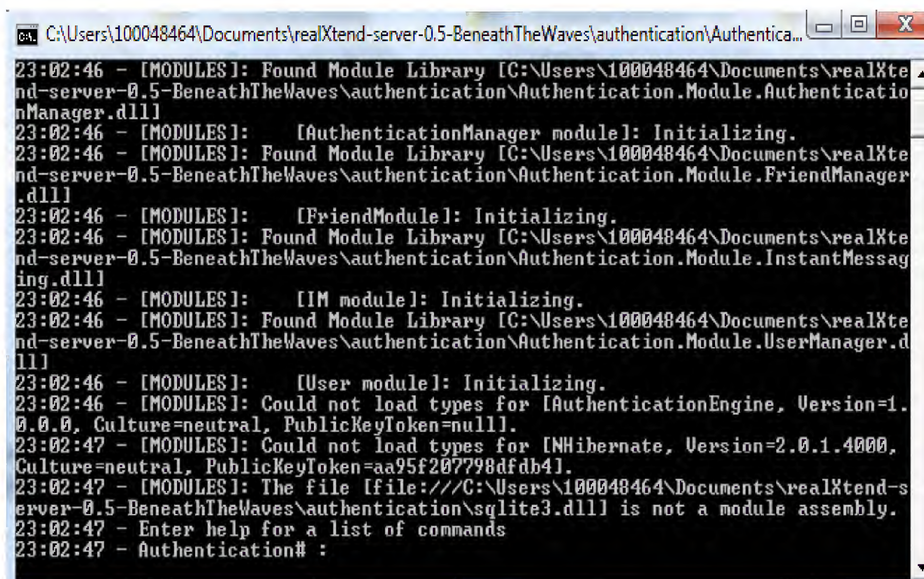
```
external_host_name="192.168.2.148"
```

- Ahora arranca todos los servidores activando StartStandaloneRexServer.bat:



7.17.5.4. Ahora tú tienes que añadir nuevos usuarios a tu servidor de autentificación

- Selecciona la consola de windows authentication.exe:



- Escribe "create user" en la ventana y presiona enter:

```
create user
```

- First name: primer nombre del avatar.

```
23:07:11 - First name:
```

- Presiona enter después de escribirlo. Esto se muestra con el apellido en la parte superior del avatar en el mundo.

```
saga
```

- Last name: el apellido del avatar.

```
23:08:48 - Last name:
```

- Presiona enter después de que lo escribas.

```
Bayn
```

- Account (cuenta): el nombre de la cuenta para utilizar al intentar acceder al servidor o al utilizar avatargenerator. Ejemplos de nombres de cuenta son rexuser, testuser, StacyB, JackBuyer o NJO.

```
23:11:43 - Account:
```

- Introducimos la cuenta:

```
100048464
```

- Real name: el nombre real del avatar

```
23:18:27 - Real name:
```

- Introducimos el Real name de nuestro avatar:

```
eduardo
```

- Password: la contraseña del usuario para conectase en el metaverso.

```
23:20:39 - Password:
```

- Introducimos la clave del avatar.

- Avatar storage url (url de almacenamiento del avatar):

```
23:24:38 - Avatar storage url
```

- Si configuraste la configuración XML correctamente, el valor predeterminado de esta debe estar bien, que usan IP externa de su máquina con algunos parámetros agregados a ella. Si el valor por defecto está bien, basta con pulsar enter. Si necesita cambiar la ip, copiar y pegar el resto de la url hasta el final.

```
(default: http://192.168.2.148:10000/avatar/9d315508146e4c5f9dfdadc858cbb5d8)
Just input the first par before /avatar/UID::
```

- Nos pide que introduzcamos el valor de la IP externa, porque el UID del avatar no se puede modificar, como vemos que el valor de la IP externa por defecto que nos muestra el programa es 192.168.2.148, que es la que posee nuestra maquina, hacemos clic en la tecla enter.

- Grid url:

```
23:28:08 - Grid url (default http://192.168.2.148:9000):
```

- Si establece la configuración del archivo xml correctamente, el valor predeterminado de esta debe estar bien, que usan IP externa de su máquina: 10000. Si el valor por defecto está bien, basta con pulsar enter.

- Vemos que la dirección IP externa que es 192.168.2.148 está correctamente, por lo tanto pulsamos la tecla enter.

- Start Region X: x-cordenada, 1000 para la localización por defecto la cual es donde la sim es normalmente localizada.

```
23:32:04 - Start Region X + (default 1000):
```

- Pulsamos la tecla enter.

- Start Region Y: y-cordenada, 1000 para la localización por defecto la cual es donde el sim es normalmente localizada.

```
23:33:29 - Start Region Y + (default 1000):
```

- Pulsamos la tecla enter.

7.18. Crear una cuenta en ReactionGrid

7.18.1. Crear una cuenta

Vamos a crear una cuenta en ReactionGrid.

- Vamos a la siguiente dirección web:

http://www.reactiongrid.com/

- Pinchamos en el menú con la opción **Main Portal**:



Buscamos en el "Main Portal", la opción "**Free User Account Registration**, y hacemos clic en ella.

- Rellenamos los datos que nos pide, y elegimos un avatar:

Get a Free ReactionGrid User Account:

First Name	giaa
Last Name	Quixote
Password	*********
Confirm Password	*********
Email	›100048464@gmail.com
SkypeID (optional)	

- Leemos las condiciones del sitio Web:

Click here to read our Terms and Conditions.

- Aceptamos los términos:

I have read the Terms and Conditions of use, and agree to comply with these. I understand that failure to adhere to these can result in suspension of my account. ☑

- Introducimos el código que nos piden, y damos al botón "Resgister Now":

Type the two words:
ing plus

reCAPTCHA™
stop spam.
read books.

Register Now

- Nos sale la siguiente pantalla:

Get a Free ReactionGrid User Account:

Registration Complete

NEW: Run the ReactionGrid Grid Configurator tool for the fastest route to get logged in to ReactionGrid! For more information, help and support on this tool head to the Configurator Help page.

For full instructions on logging in and client downloads, head here to the support section of the ReactionGrid web site.
If you already have the Hippo client installed (available here) you can use the login URI of **http://reactiongrid.com:8008** to add ReactionGrid to the list so you can log in.

HOW TO CONNECT TO REACTION GRID:

* Download and install the Hippo viewer
* Run the program
* On the splash screen, click the GRIDS button
* On the pop-up window, click the GRIDS tab
* At the top of the box, click ADD (we are adding Reaction Grid)
* For LOGIN URI enter: http://reactiongrid.com:8008/
* For PLATFORM choose: OpenSim
* For GRID NICKNAME enter: reactiongrid
* For GRID NAME enter: ReactionGrid
* Click the DEFAULT button at the top of the box
* Click the OK button

You're now ready to enter your user name and password.
Click here for a video on how to login. Use "High Quality".

7.18.2. Configurar ReactionGrid

Vamos a configurar el explorador para que funcione ReactionGrid.

- Miramos nuestro correo electrónico, y vemos un correo electrónico:

sendmail » **ReactionGrid Registration Confirmation** - Thanks for registering an account with ReactionGrid - the 3D Virtual Worlc

- Nos han enviado el siguiente mensaje:

ReactionGrid Registration Confirmation Recibidos | X

sendmail@reactiongrid.com para usuario mostrar detalles 12:50 (38 minutos antes) Responder

Thanks for registering an account with ReactionGrid - the 3D Virtual World for business, education, collaboration, and learning.

You registered the avatar name: giaa Quixote

If you forget your password, head to http://www.reactiongrid.com/support/resetpassword.aspx.

Your avatar should appear as you selected when you first login, but all clothing options from the registration form will appear in your inventory, and there's even more available for free on Welcome sim if you need to change your appearance.

For instructions on how to log in to ReactionGrid, please head to http://www.reactiongrid.com/Support/LoggingIn.aspx

We look forward to seeing you on the grid!

Kyle, Robin and Chris.

- Hacemos clic en **ReactionGrid Grid Configurator** y nos muestra la siguiente pantalla:

Welcome to the ReactionGrid Client Configurator

The following prerequisites are required:

- .NET Framework 2.0 (x86)

Click the button below to install any prerequisites and run the application.



A zip file containing the application .exe is available for download here.

- Hacemos clic en el enlace **here**, y nos descargamos el siguiente programa:


ReactionGridConfigurator.zip
312 KB

- Descargamos el **Hippo viewer**:

  http://forge.opensimulator.org/gf/project/opensim-viewer/frs/

- Arrancamos el "Hippo viewer":


Hippo ReactionGrid
Acceso directo
1 KB

- Sobre la pantalla de acogida, pulse el botón de **Grids**:



También puedes seguir la siguiente ruta: "Edit / Preferences"

- Sobre la ventana emergente, pulse la etiqueta de **Grids**:



- Haz clic en el botón **Add** (añadimos "ReactionGrid"):



- En **Login URL** introduce: http://reactiongrid.com:8008/

Login URI: http://reactiongrid.com:8008/

- Hacemos clic en **Get Grid Info**:



- Para PLATFORM escoge: OpenSim

Platform: OpenSim

- Para GRID NICKNAME introduce: reactiongrid

Grid Nickname: reactiongrid

- Para GRID NAME introduce: ReactionGrid

Grid Name: ReactionGrid

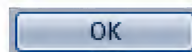
- Haz clic en el botón **Apply**:



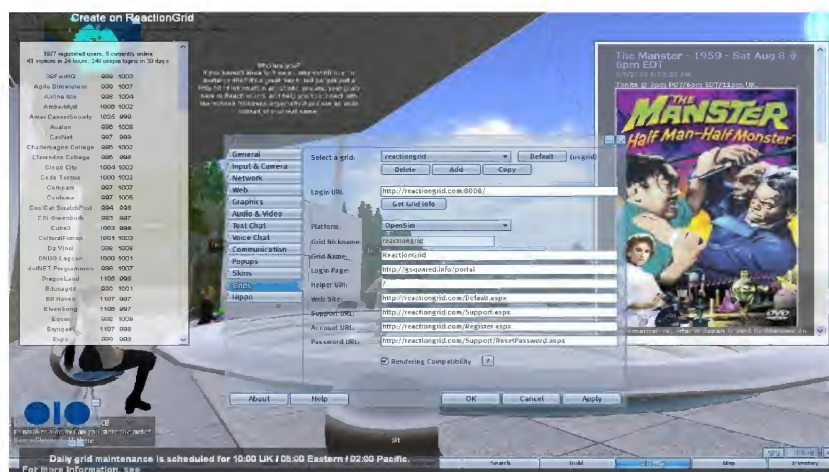
- Haz clic en el botón **Default** en lo alto de la caja:



- Haz clic en el botón **OK**:



- Ya nos hemos conectado a ReactionGrid:



7.19. Sloodle en ReactionGrid

Vamos a buscar las herramientas de Sloodle en el mundo virtual ReactionGrid.

- Al conectarnos a ReactionGrid, nos sale un listado ordenado por orden alfabético la lista de las regiones que hay en ReactionGrid, buscamos la región de Sloodle.

- Hacemos clic en "Sloodle 1000 997":



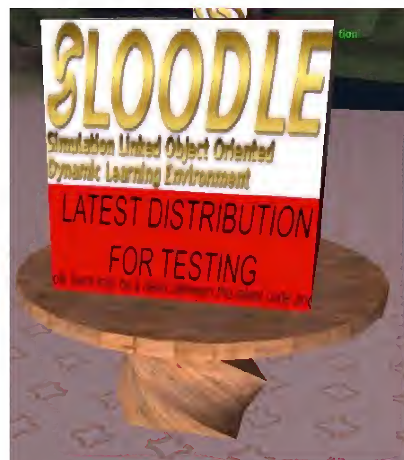
- Introducimos nuestros datos:

First Name: giaa
Last Name: Quixote
Password: *********
Log In
Start Location: Sloodle
Remember password

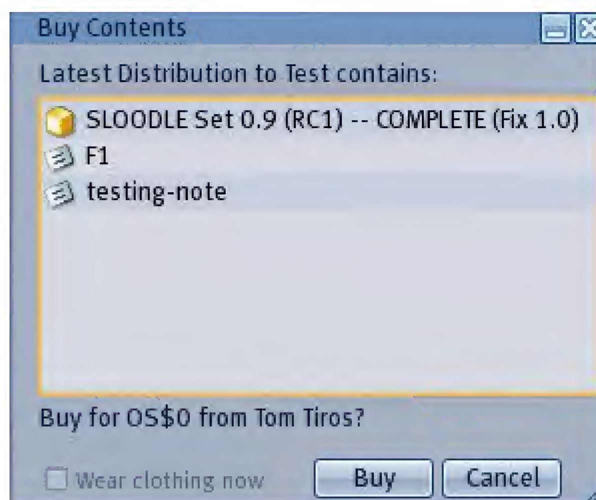
- Ya estamos en Sloodle de ReactionGrid:



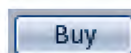
- Hacemos clic en el cuadro siguiente:



- Nos sale el siguiente cuadro de dialogo:



- Damos al botón **Buy**:



- Hacemos clic en el inventario:



- Nos metemos en la pestaña **Recent Items**:

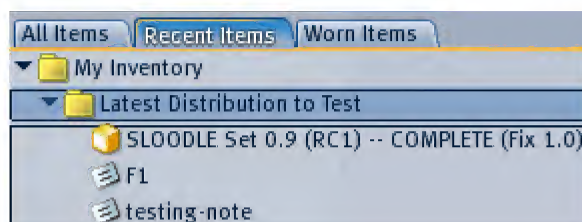


Figura 196. Recent Items Reaction Scripts.

Vemos que se nos ha creado una carpeta llamada **Lastest Distribution to Test**.

Conclusiones OpenSim

Hemos creado nuestra propia Grid de OpenSim, lo cual nos posibilita tener todas las regiones que queramos de forma gratuita pues nuestro metaverso es mantenido por varios ordenadores y también nos posibilita poder crear toda clase de infraestructuras pues no nos cuesta dinero subir materiales a nuestra Sim para crear diversos objetos, recordemos que Second Life nos cobra cada vez que queremos subir un material (dibujo, textura, …). Hemos cargado las herramientas de Sloodle en OpenSim, lo cual nos posibilita poder impartir clases en nuestras Sim, y nuestro metaverso esta mantenido por varios ordenadores de tal forma que la creación de un entorno educativo es totalmente viable.

CONCLUSIONES

Conclusiones

El proyecto tiene un gran potencial, al haberlo desarrollado de manera totalmente gratuita, permitiendo crear un grid y utilizar hypergrid mediante OpenSim, y damos la posibilidad de vigilar el metaverso creado mediante bots.

El grid nos permite tener todas las islas que queramos y unir las Sim de otras entidades a nuestro grid. De esta manera podemos ofrecer Sim gratuitas a otras entidades que quieran utilizar nuestros recursos.

Nos posibilita crear un grid en el cual solo formen parten Sim de universidades y los avatares que entren al grid sean únicamente alumnos de las distintas universidades.

Al tener islas ilimitadas, cada clase que se imparta en la universidad puede tener su propio Sim para impartir sus clases.

El hypergrid tiene un potencial tremendo, nos permite hacer enlaces a otros grid, metaversos o standalone, y de esta manera posibilitamos el poder teletransportarnos a otras Sim sin necesidad de que estén en nuestra grid.

En cuanto a la educación en nuestro metaverso, tenemos la posibilidad de crear en cada Sim un set Sloodle, y de esta manera poder impartir clase en cada clase. Los set de Sloodle continuamente se están actualizando, pues están desarrollando nuevas herramientas para dar clase.

Moodle es una estupenda herramienta gratuita, que tiene un gran potencial, por su sencillez de uso, y por las funciones que nos permite realizar.

OpenSim continuamente están desarrollándolo, y actualizándolo, y es completamente configurable.

Futuras líneas/trabajos

Las futuras ampliaciones son las siguientes:

- Investigación de la tecnología RealXtend para la creación de metaversos gratuitos.
- El bot responde a más preguntas.
- El bot acate más órdenes. De esta manera se podrían utilizar en un lugar de forma más eficiente para que pudieran ayudar.
- Utilizar Linux para utilizar OpenSim, y de esta manera no se tendría que pagar ninguna clase de licencia.
- Utilizar OpenSim para crear todas las islas necesarias con todos los edificios necesarios para poder construir con todos los detalles el grid de la universidad Carlos III de Madrid y así poder empezar a impartir clase a través de internet.
- Investigación de la comunicación de los prims del metaverso con java y php.

GLOSARIO

Glosario

ACCESS CHECKER DOOR	*Esta utilidad, realiza las mismas funciones que Sloodle Access Checker, pero en vez de activarse, al entrar el avatar en una zona, se activa cuando el avatar toca el objeto. Si al avatar se le permite entrar, se abre para dejarlos pasar.*
ACTIVIDADES	*Bloque que contiene una lista de todas las actividades existentes en el curso o asignatura, agrupadas por tipo de actividad.*
AVATAR	*Es la representación gráfica de un usuario.*
BASE DE DATOS	*En el caso de Moodle esta opción ofrece información de manera general de todos y cada uno de los participantes del curso.*
CALIFICACIONES	*En esta sección se encuentra una lista de todas las actividades calificables, junto con la puntuación obtenida hasta el momento. Esta lista es privada, cada alumno sólo puede ver sus propias calificaciones.*
CAMPING	*Actividad que permite ganar L$ Linden Dólar, por bailar, sentarse o realizar alguna actividad aumentando el tráfico del lugar.*
CHAT	*Escribir en público.*
CHOICE HORIZONTAL	*Encuentas, elecciones o recopilación de información.*
CONSULTA	*Opinión acerca de algo.*
CUESTIONARIO	*Lista de preguntas.*
ENCUESTA	*Conjunto de preguntas típicas.*
GRID	*Es una colección de sims (islas).*
HOST	*Es un servidor, usualmente llamado simulador (sim).*
HUD	*Objetos que ocupan la pantalla de la aplicación, permitiendo realizar funciones y configuraciones.*
IM	*Instant Message (Mensaje instantáneo)*
INVENTARIO	*Es un escaparate personal donde podemos guardar todos nuestros artículos. Objetos, ropa, mensajes, etc.*
LANDMARK	*Dirección que señala algún punto de ubicación específico.*

LINDEN DOLAR	*Second Life tiene su propia economía y moneda conocida como dólares Linden (L$), que es usada por los residentes para comprar y vender los artículos y servicios creados dentro del mundo virtual.*
LSL	*Es un lenguaje de programación creado por Linden Labs.*
LOGIN ZONE	*Registro de avatares.*
MACHINIMA	*La creación de películas utilizando motores de videojuegos.*
METAVERSO	*Los metaversos son entornos donde los humanos interactúan social y económicamente como iconos a través de un soporte lógico en un ciberespacio que se actúa como una metáfora del mundo real, pero sin las limitaciones físicas.*
METAGLOSS	*Accedemos a los glosarios de Moodle desde el chat SL.*
NEWBIE	*Es un residente nuevo en SL.*
PARCEL	*Es un área de tierra que pertenece a un usuario o grupo.*
PASSWORD RESET	*Restablecimiento de contraseña.*
PRESENTER	*Presentaciones multimedias.*
PRIM	*Es un objeto.*
PRIM DROP	*Permite a los estudiantes enviar trabajos.*
QUIZ CHAIR	*Realizar cuestionarios de un curso desde SL.*
QUIZ PILE ON	Realizar cuestionarios multiusuario de un curso de SL.
REGENROL_ BOOTH	*Permite a un usuario registrado en un sitio Moodle, registrar su avatar en un sitio de Sloodle.*
RESIDENTE	*Cada persona con acceso al mundo virtual.*
REZEAR	*Crear un objeto en un simulador de Second Life, bien sea desde el inventario o desde la herramienta de construcción.*
RL	*Vida Real.*

SANDBOX	*Es un sitio especial para la creación y tratamiento de objetos.*
SEARCH	*Divididas por personas, grupos, clasificados, lugares y eventos.*
SHAPE	*En Second Life cada avatar tiene una forma definida y modificable.*
SIM	*Simulador. Es una isla virtual.*
SKIN	*Piel.*
SL	*Second Life. Mundo Virtual en 3D creado por Linden Lab.*
SLurl	*Dirección de referencia de algún lugar en Second Life.*
TELEPORT	*Método de transporte entre un punto y otro en Second Life.*
TEXTURE	*Es una imágen aplicada a la superficie de un objeto.*
TOOLBAR_ GIVER	*Para gestionar el avatar y el blog desde Second Life.*
UUID	*Es un identificador único y global.*
VECTOR	*Es un vector o matriz de punto flotante usado para representar posiciones, colores, etc.*
VENDING MACHINE	*Sirve para distribuir objetos a tus estudiantes.*
WEBINTERCOM	*Nos sirve para interconectar nuestro chat de Moodle, con nuestro chat de Second Life.*

BIBLIOGRAFÍA

Bibliografía

[1] Página Web de Second Life:

https://secure-web39.secondlife.com/my/account/login.php?lang=en&type=second-life-member&nextpage=/my/index.php?lang=en

[2] Crear una cuenta en Second Life:

http://www.virtual-spain.es/

[3] Requerimientos de Second Life:

https://secure-web45.secondlife.com/my/support/system-requirements/

[4] Utilidad Scratch:

http://web.mit.edu/~eric_r/Public/S4SL/

[5] Página web de Moodle:

http://moodle.org/

[6] Pagina web de moodlefarm: Prueba gratuita de host gratuito con Moodle y Sloodle:

http://moodlefarm.socialminds.jp/

[7] Página web keytoschool: Prueba gratuita de host gratuito con Moodle y sloodle:

http://www.keytoschool.com/

[8] Descargar programa Cron:

http://docs.moodle.org/es/Cron

[9] Información de Email_authentication en la página oficial de Moodle:

http://docs.moodle.org/es/Email_authentication

[10] Manual de consulta del profesorado: Jesús Baños Sancho.

http://aulavirtual2.educa.madrid.org/mod/resource/view.php?inpopup=true&id=5659

[11] Página de OpenSimulator:

http://opensimulator.org/wiki/Main_Page

[12] Guía para crear un OpenSim Propio usando binarios compilados y cómo crear nuevas Regiones:

http://sombra.lamatriz.org/wiki/index.php/Guia_para_crear_un_OpenSim_Propio_usando_binarios

[13] Instrucciones para crear un OpenSim propio sobre Windows XP:

http://sombra.lamatriz.org/wiki/index.php?title=Guia_para_crear_un_OpenSim_Propio&rcid=234

[14] Portal la comunidad: Grupo la Cripta: Darok Kaminski, Leizah Rosana, Silvia Mills Alesandra Capra y Albert Revolution.

http://sombra.lamatriz.org/wiki/index.php/Indice_del_Portal_de_la_Comunidad

[15] Guía para conectar un grid al Sim:

http://sombra.lamatriz.org/wiki/index.php/Guia_conectar_un_sim_al_grid

[16] Binarios de OpenSim:

http://builds.opensimulator.org/

[17] UUIDs ó identificadores únicos de estas regiones:

http://www.famkruithof.net/uuid/uuidgen

[18] Comandos para OpenSim:

http://opensimulator.org/wiki/Server_Commands

[19] Página principal de OsGrid:

http://www.osgrid.org/elgg/

[20] Instrucciones de OsGrid:

http://osgrid.org/docs/instructions.htm

[21] Metaverso OpenSim en GNU/Linux:

http://nonumerable.net/metaverso-opensim-en-gnulinux/

[22] Página de ReactionGrid:

http://www.reactiongrid.com/

[23] Login dentro de ReactionGrid:

http://www.reactiongrid.com/Support/LoggingIn.aspx

[24] Instrucciones de cómo logearse en ReactionGrid:

http://www.reactiongrid.com/Support/LoggingIn.aspx

[25] FAQ de OpenSim:

http://opensimulator.org/wiki/EspriFAQ

http://opensimulator.org/wiki/Configuration

http://opensimulator.org/wiki/Configuring_Regions

http://opensimulator.org/wiki/Mysql-config

[26] Enlazar regiones mediante hypergrid:

http://lab40.homelinux.net/wiki/doku.php?id=opensim:como_enlazar_rexions_externas_doutros_servers

[27] Página Web de Sloodle:

http://www.sloodle.org/moodle/

[28] Documentación de Sloodle:

http://slisweb.sjsu.edu/sl/index.php/SloodleUserDocs

http://www.sloodle.org/blog/?page_id=33

[29] Pagina web de LibOpenMetaverse:

http://www.openmetaverse.org/projects/libopenmetaverse

[30] Wiki LibOpenMetaverse:

http://lib.openmetaverse.org/wiki/Developer_Portal

[31] Guía para crear un bot básico:

http://lib.openmetaverse.org/wiki/How_to_create_a_basic_libopenmv_bot%3F

[32] Análisis y diseño estructurado y orientado a objetos de sistemas informáticos. Editorial MC Graw Hill.

[33] Wiki Second Life:

http://wiki.secondlife.com/wiki/Main_Page

[34] LSL:

http://wiki.secondlife.com/wiki/LSL_Portal

[35] LipSync:

http://wiki.secondlife.com/wiki/LipSync

[36] Libros de Moodle:

http://docs.moodle.org/es/Manuales_de_Moodle

[37] Documentación de sloodle:

http://slisweb.sjsu.edu/sl/index.php/SloodleUserDocs

[38] Foro LibOpenMetaverse:

http://libsecondlife.13613.n2.nabble.com/

[39] Wiki libopenmetaverse:

http://lib.openmetaverse.org/wiki/Developer_Portal

[40] Mike McKay:

https://www.sloodle.org/blog/?page_id=33#SLOODLE1.x

http://www.youtube.com/SLOODLE

[41] Speech Recognition en Windows vista:

http://msdn.microsoft.com/en-us/magazine/cc163663.aspx

[42] pikkuBot:

http://www.pikkubot.de/dokuwiki/doku.php?id=installation:help

[43] botAriel:

http://www.sustentum.com/sustentum/ariel.asp

[44] Albert Revolution:

http://exploradorvirtual.blogspot.com/search?updated-max=2010-02-13T20%3A32%3A00%2B01%3A00&max-results=7

http://exploradorvirtual.blogspot.com/

[45] Hispagrid:

http://foro.hispagrid.com/viewforum.php?f=5

[46] Realxtend:

http://wiki.realxtend.org/index.php/Main_Page

[47] Marlonj:

http://www.marlonj.com/blog/author/marlonj/page/150/

[48] Irene Muni:

http://www.irenemuni.com/foro/opensim-vf14.html

[49] Atlante Guerrero:

http://www.creasl.org/

[50] Openlife:

http://openlifegrid.com/

[51] Wonderland:

http://www.wonderlandblog.com/

[52] Croquet

http://www.opencroquet.org/index.php/Main_Page

http://el-directorio.org/Croquet/Manual

[53] Open Cobalt:

http://www.opencobalt.org/

[54] Grid 3rd rock grid:

http://site.3rdrockgrid.com/

[55] New World:

http://www.newworldgrid.com/lang/en-us/

[56] Open Neuland y Wilder Westen:

http://www.talentraspel.com/grid_open_neuland.html

http://www.talentraspel.com/grid_wilder_westen.html

[57] Virtyou:

http://virtyou.com/en/

[58] Your alternative life:

http://www.youralternativelife.com/

[59] Camtasia Studio:

http://www.techsmith.com/camtasia.asp

[60] Requerimientos software y hardware para Second Life:

http://secondlife.com/support/system-requirements/

[61] Explorador Hippo Viewers:

http://www.osgrid.org/elgg/pg/utilities/software

[62] Realxtend:

http://www.realxtend.org

[63] Poser:

http://my.smithmicro.com/shop/graphics.html

[64] 3D Space Navegator:

http://www.3dconnexion.com/products/spacenavigator.html

[65] Voice en OpenSim:

http://opensimulator.org/wiki/Freeswitch_Module

[66] Virtus:

http://www.association-virtus.org/